POLITECNICO DI TORINO

MASTER THESIS

Integrating New Data into Generative Models of Biomolecular Sequences

Author:
Giovanni PEINETTI

Supervisors:
Andrea PAGNANI
Martin WEIGT

A thesis submitted in fulfillment of the requirements for the Master degree in

Physics of Complex Systems

March 27, 2024

Acknowledgements

First of all, I would like to thank my Martin and Francesco. I learnt so many things from them during these months and I will always remember their kindness. I felt like I was at home, they were always there for an advice and they really cared about me and my work. Without them I would not have gone far.
Secondly, I surely have to thank Andrea for making all of this possible and for being available at every time, your advice was really precious.
Then a huge thanks goes to Alya, Roberto, Lorenzo and again Francesco for being good friends when I was in Paris. I will always remember this months with happiness and a bit of nostalgia.
I cannot forget to mention my classmates Matteo, Alessandro, Salvatore and Michele. If I have completed this master it is also thanks to them who supported me all along the way.
I must thank all my friends, especially Simone, Loris and Stefano for always being there for me.
Furthermore, I want to express my gratitude for my family that constantly supported me even when we had bigger problems and also when I was far away.
Last but not least, my biggest thanks goes to Alessia. Her love and support gave me a lot of strength and confidence. I could have not done this without her, I am so happy that our roads crossed again.

Abstract

The design of functional artificial biomolecules has been one of the main interests of biotechnology in recent years. The aim is to design sequences that have the same functionality of the natural ones and comparable features. Data-driven approaches are one of the more successful strategies.
In Machine Learning generative statistical models are tools to generate artificial biomolecular sequences. They are trained on Multiple Sequence Alignments of homologous families which consist of positive unlabelled sequences. In literature there are several examples where generative models have been built successfully to generate functional RNA and Proteins.
Relying on maximum entropy principle, Direct Coupling Analysis (DCA) models are based on the Boltzmann Distribution in physics. They are built learning a Potts model from data via Maximum Likelihood and they can be used to sample artificial sequences.
Now thanks to the advent of new quantitative high-throughput experiments, more and more quantitatively annotated sequences emerge. This abundance of information presents unprecedented opportunities to improve generative models, significantly enhancing their accuracy and efficacy in synthetic biology.
Using the framework of energy-based models, in this thesis a new statistical-physics inspired algorithm was developed to integrate these labelled data into the construction of a better generative model. A new objective function was designed to include the information from both the unlabelled and labelled data. Its maximisation is equivalent to adjust the target frequencies for the training and no back-propagation is needed: it can be thought as a refinement of the original generative model.
The goal of this feedback system is twofold: to minimise the production of non-functional sequences and to engineer new artificial sequences that exhibit specific desired characteristics, such as structural compatibility. Our algorithm was applied to train models both on synthetic and real data and it provided exceedingly good results in directing the generation towards the desired features. To validate our techniques, a series of biological experiments is scheduled in the near future.

Contents

List of Abbreviations

DCA	**D**irect **C**oupling **A**nalysis
bmDCA	**B**oltzmann **M**achine **D**irect **C**oupling **A**nalysis
eaDCA	**E**dge **A**ctivation **D**irect **C**oupling **A**nalysis
MCMC	**M**arkov **C**hain **M**onte **C**arlo
NAT	**NAT**ural sequences
ART	**ART**ificial sequences
MEP	**M**aximum **E**ntropy**P**rinciple
LLM	**L**arge **L**anguage **M**odel
IID	**I**dentically **I**ndependently **D**istributed

Chapter 1

Introduction and Biology Concepts

The aim of generative models is to design sequences that have the same functionality of the natural ones and comparable features.
They have been used for protein design [1] and artificially generated sequences were used to effectively substitute natural ones in microorganisms [2].
Generative models of sequences are typically trained in an unsupervised way on non-annotated sequence data, as presented by MSA of protein or RNA families.
In recent years there is an increasing possibility of performing new quantitative high-throughput experiments and more and more quantitatively annotated sequences are available. This abundant experimental information presents unprecedented opportunities to improve generative models[3].
The goal of this thesis is to present methods for integrating annotated data into existent generative models of biomolecules.
This chapter aims at presenting the biology concepts and settings that will be used in the following of the thesis.
The focus will be then restricted on methods to build relatable generative models of biomolecular sequences: Direct Coupling Analysis models and the ViennaRNA package (Ch. 2).
Afterward, starting from a previous take on the reintegration of experimental data into generative models, the aim is to show how to actively go further in this direction (Ch. 3) and how the new reintegration methods can be applied to both synthetic data (Ch. 4) and real data (Ch. 5) with promising results.

1.1 A Brief Introduction to RNA

Ribonucleic acid (RNA) is a polymeric biomolecule which plays a fundamental role in many biological processes.
Each RNA is composed of nucleotide units linked by phosphodiester bonds and it can fold into hairpins, pseudo-knots, riboswitches and a lot more shapes.
A correct folding is crucial for the molecule to function and the nucleotide sequence is often not enough.
The three-dimensional structure of RNA is deeply connected with the functionality of the sequence [4]. The building blocks of RNA molecules are the nucleotides: Adenine, Guanine, Cytosine and Uracil (instead of Thymine).

As in DNA, they can form Watson-Creek pairings (Cytosine-Guanine and

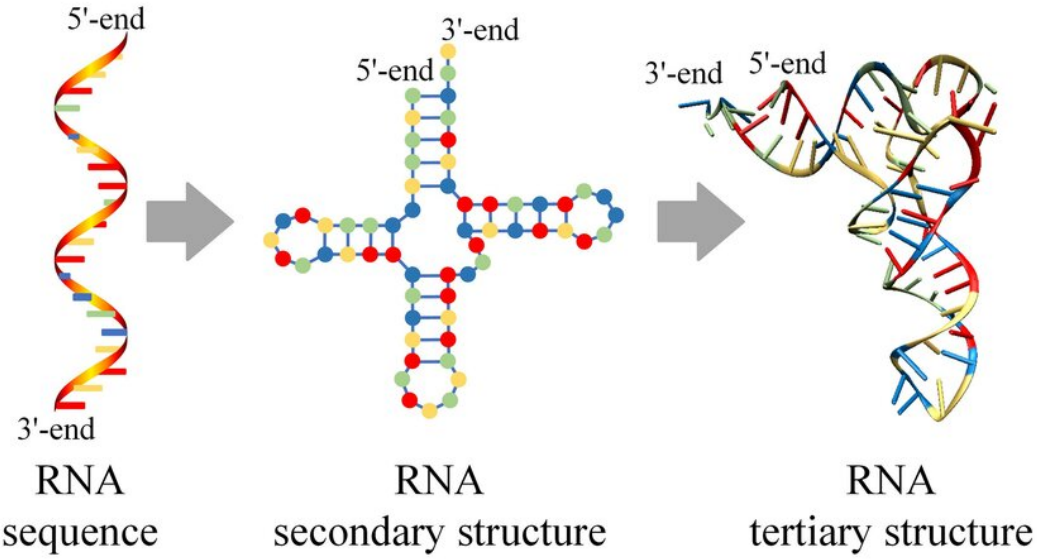

FIGURE 1.1: Representation of the hierarchical level of structural organisation in RNA

Adenine-Uracil). Alongside the canonical base-pairing there is also the possibility that a wobble base-pair between Adenine and Guanine is formed. These types of bonds are less stable.
There are three different levels of RNA structure. The primary structure is identified by the sequence of nucleotides.
The secondary structure is determined by the intra-chain base-pairing. It divides the sequence in domains and it is two-dimensional. The key structural elements are helices and loops. Helices are regions where bases are paired through hydrogen bonds, forming stable double-stranded structures. Loops instead are unpaired regions in RNA that connect helical domains.
The secondary structure of RNA is very important for functionality as it determines how RNA interacts with other biomolecules and how it performs specific tasks within the cell.
The tertiary structure is instead driven by the interaction of the various domains and it is three-dimensional.
A pictorial view of RNA hierarchical strucure is given in Figure 1.1.
In this thesis we will talk about the secondary structure but mostly a RNA molecule will be seen as a sequence made of nucleotides, a word in a 4 letter alphabet.
It is important to underline the fact that the sequence space is very huge : this also justifies the need of statistical generative models.
For a RNA molecule which is 200 nucleotides long, since each nucleotide has four possible options, the sequence space is made of $4^{200} \sim 2.6 \times 10^{120}$ possible sequences. This number is astronomically large : the estimated number of atoms in the known universe is 10^{78}. A computer-assisted statistical approach is needed to explore this huge space: Machine Learning techniques are designed exactly to do this.

Those techniques are very data-hungry but, recently, the sequencing technology has improved a lot and a huge number of sequences is accumulating at exponential speed.
RNA sequence data is collected in families [5]. An RNA homologous family groups together molecules which have similar structure, similar function and come from the same evolutionary branch (homologous sequences). They are unlabelled collections and they can be considered as variants of the same sequence.
These families are available in the Rfam database and there are approximatively 4000 families. Each family is made of order of thousands sequences.
The effect of evolution is not only to modify the type of nucleotide (mutations) but also the length of sequences can change with a variability of more than ± 20 nucleotides. This variable length is a huge problem. Thankfully there are lot of pre-existent tools based on Infernal covariance models [6] which account quite thoroughly for this problem and so in the following we will not deal with the "sequence alignment" problem [7] also because in Rfam dataset all the families are already aligned.

1.2 Azoarcus Setting

Abiogenesis refers to the process through which the simplest forms of life originated from non-living matter. The "replication first" is one of the most successful approach : it highlights the role of replicase ribozymes, polymers that can self-replicate. Self-reproduction is the ability to generate copies of oneself. Naturally occurring self-reproducing RNAs have not been discovered but the method of in vitro evolution made the development of RNA replicases possible. The Azoarcus bacterium's group I intron ribozyme, in the followong shortly named "Azoarcus" as done in the community, consists of 197 nucleotides and it has been modified for self-reproduction. Its structure is divided in 4 segments, as shown in Figure 1.3
Azoarcus can be fairly considered the only known self-replicator. Furthermore it belongs to the family of Group I introns which are capable of self-splicing : they excise themselves from messenger RNAs. A schematic representation of this phenomenon is provided in Figure 1.2.
Is RNA reproduction widespread in the sequence space? If so, the plausibility of the origins of life in the RNA world context (RNA is considered as the predecessor of DNA and proteins) would increase greatly.
RNA world hypothesis [8][9] requires self-replicators to emerge spontaneously. If it were a widespread phenomenon instead of restricted to a single sequence, this would support the hypothesis.
To find potential self-replicator candidates, a sequence alignment was produced based on Azoarcus and other Group one intron RNAs : it is an alignment of poor quality, very gapped. Despite this, it is possible to learn models on it and generate artificial mutations that can be tested. However, due to the bad alignment quality, the models are not expected to be very accurate, and therefore fail to produce functional mutant sequences after few mutations.

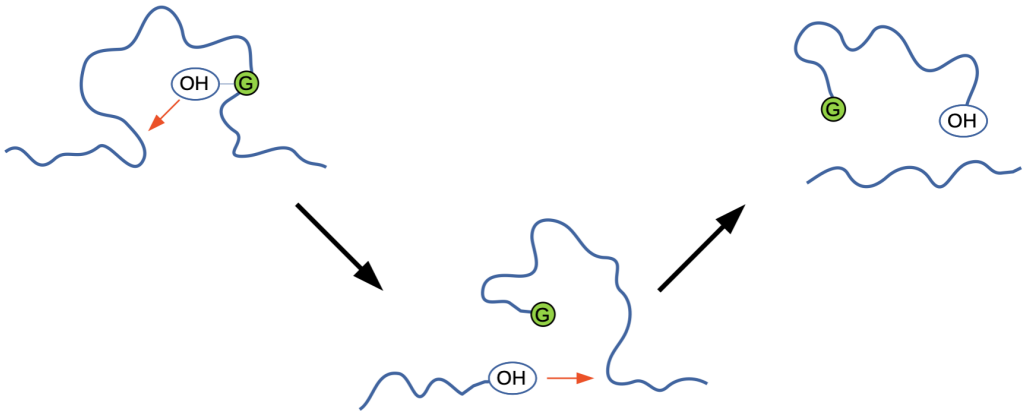

FIGURE 1.2: Steps of the phenomenon of self-splicing

The work proposed in this thesis is linked to this research framework and Chapter 5 is an extension of this.
These mutations can be produced artificially: RNA design is possible thanks to sophisticated biological procedures such as rational design, directed evolution, and in silico design.
Artificial sequences can then be mass-tested for self-splicing in high-troughput experiments. This is a good proxy for self-replication. On top of this there are also low-throughput experiments for self-replication.
The aim is not only to find new self-replicants but also to try to bound how many functioning self-replicators are present in the huge sequence space.
One of the goals of this thesis is to start from the statistical models (DCA models) that have been applied until now and, thanks to the experimentally tested sequences in the Azoarcus setting, build new refined models that outperforms them.
This not only helps the cause of RNA world theory but also underlines the importance of the experiments in this field.

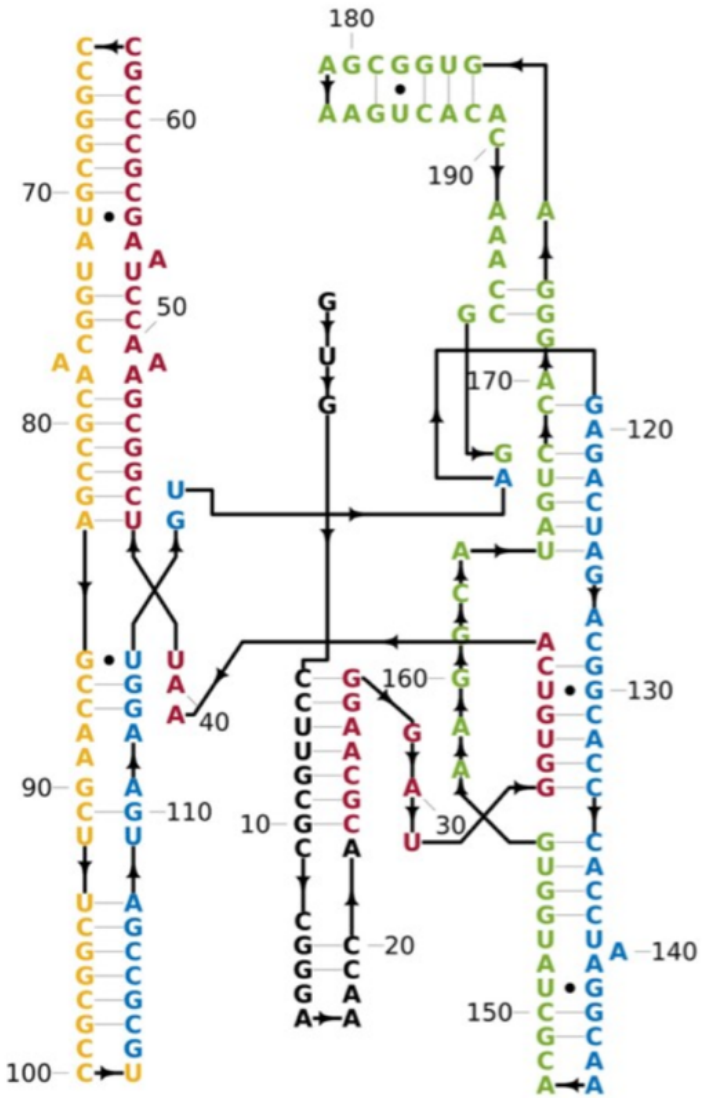

FIGURE 1.3: Structure of the Azoarcus bacterium's group I intron ribozyme: Segments are colored differently : if they are provided separately, the RNA molecule is able to catalyze the assembly oif full length RNAs

Chapter 2

Methods

2.1 Generative Models Of Biological Sequences

As discussed in the introduction, in recent years complete genomes have been sequenced [10] and it is now possible to consider entire families of homologous sequences. All of these sequences have been observed in nature and therefore we refer to them as "natural".
Homologous families are organised in MSAs, multiple sequence alignments [11]. They can be regarded as matrices $A = \{a_i^\mu, i = 1, ..., L, \mu = 1, ..., M\}$ which contain M sequences aligned over L positions. Each entry a_i^μ of the matrix belongs to some alphabet: nucleotides for RNA, amino acids for proteins.
To these alphabets is also added the alignment gap $" - "$ which accounts for amino-acid insertions or deletions in some sequences.
In the RNA case an alphabet with 5 letters is needed while in the case of Proteins the alphabet has 21 letters.
Each row of A corresponds to a single sequence and each column to a specific site in the sequences as shown in Figure 2.1.
The aim of generative models is to generate artificial sequences that replicate

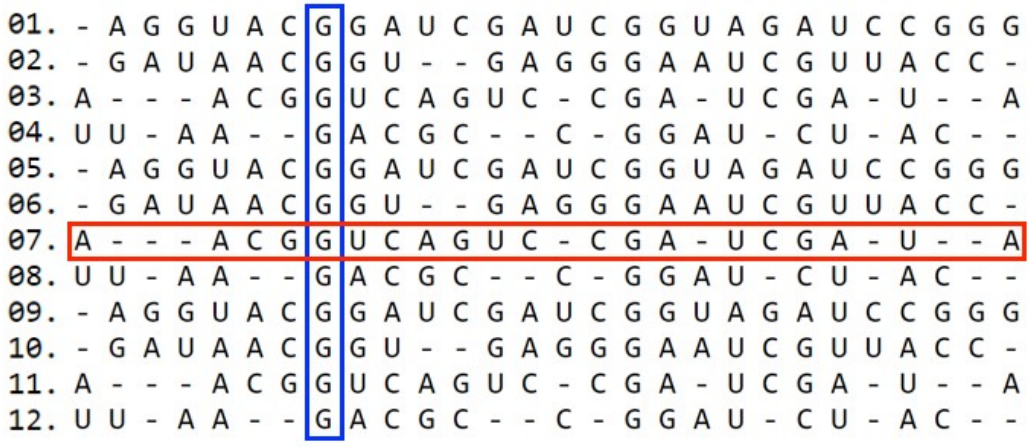

FIGURE 2.1: example of an RNA homologous family: a row corresponds to a sequence and a column represents a site across all the family. Figure courtesy of F. Calvanese

the important features of the natural ones assuming that biological information is hidden in the statistical proprieties of the data.
The underlying assumption is that, for each homologous family, there exists a probability distribution $P_0(a_1, a_2, \dots a_L)$ from which all the sequences are drawn independently.
A probabilistic generative model is built from the natural data trying to infer an approximation of the probability distribution $P_0(a_1, a_2, \dots, a_L)$ that did generate them. So an approximated probability distribution $P_1(a_1, a_2, \dots, a_L)$ is obtained and it is possible to sample artificial sequences from it. This is shown in Figure 2.2
The sampled sequences should replicate the features of the natural ones;

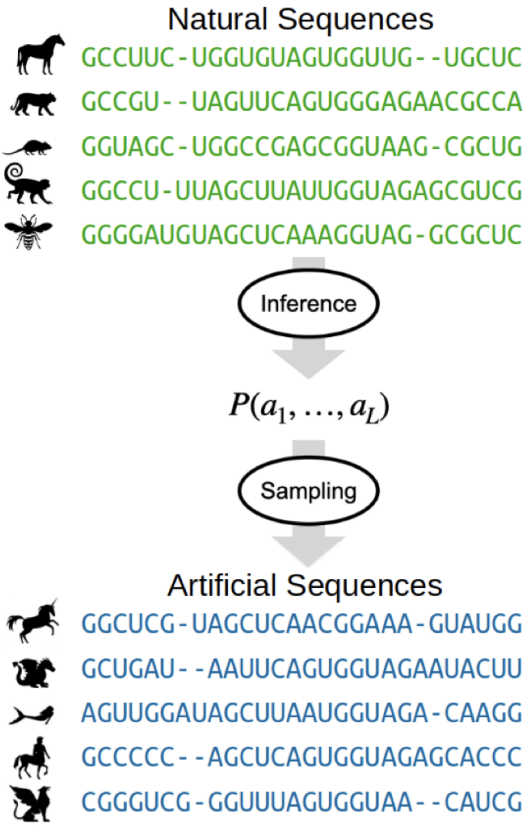

FIGURE 2.2: Functioning of a generative model. Figure reproduced from [12]

among the statistical features that one can hope to replicate there are the one-point and two-points frequencies:

- One-point frequencies $f_i(a_i)$ are the frequencies of nucleotide a_i appearing in site i
- Two-points frequencies $f_{ij}(a_i, a_j)$ are the frequencies of the pair (a_i, a_j) appearing in site i and j respectively

One point frequencies $f_i(a_i)$ account for the evolutionary phenomenon of site conservation. It is very common that in a specific site almost only one

nucleotide is observed: this can be a specific site which is directly involved in the functionality of the molecule.
For example, a specific nucleotide needs to be present in the active site of a RNA molecule or in an exposed position at the RNA surface. Then a mutation changing the nucleotide on this site will be deleterious and the site is said to be conserved.
When a nucleotide is conserved the value of $f_i(a_i)$ is informative because it will be near to the unity for a nucleotide type and very small for all the others.
Two points frequencies $f_{ij}(a_i, a_j)$ instead account for the evolutionary phe-

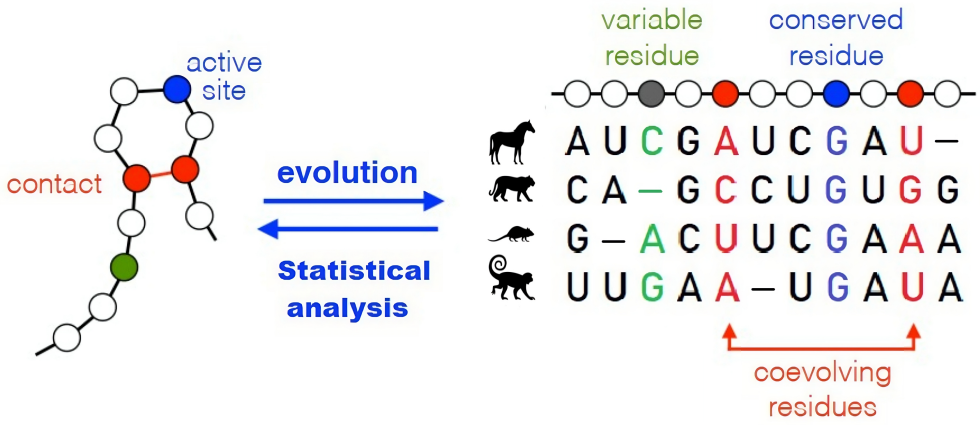

FIGURE 2.3: Visual representation of the meaning of one and two points frequencies. Figure reproduced from [3]

nomenon of co-evolution: nucleotides at different positions do not evolve independently. If a pair of sites is in contact in the folded state of the molecule and the nucleotide in one site changes due to a mutation, also the other nucleotide will likely change to maintain the ability to form the bond. In fact it is known that the preservation of the structure/function is fundamental in the process of evolution.
Looking across the family the nucleotides corresponding to these two sites will appear often as a complementary pair and this impacts on the two point frequencies $f_{ij}(a_i, a_j)$.
These concepts are represented in Figure 2.3. For sure these two quantities do not correspond to all the statistical information contained in the RNA family because there are also all the higher order frequencies.
For example one could want to replicate the three points frequency $f_{ijk}(a_i, a_j, a_k)$. These quantities consist of $5 \times 5 \times 5 = 125$ entries in the RNA case and $21 \times 21 \times 21 = 9261$ in the Proteins case. A database of homologous families is of the order of thousands of sequences and one could think it is not big enough to offer a fair representation of three sites interactions. In general the reproducing pair frequencies actually, in applications to real MSA, also reproduces the three-point frequencies to a large amount .
Models that try to fit higher order frequencies will most likely be fitting the

noise in the training data since there would be too many parameters to infer: this is referred as overfitting in Machine Learning jargon.
In literature there are several examples suggesting that one and two point frequencies are sufficient to obtain functional artificial sequences [13][14][15].
Then it will be enough to just concentrate on the first two orders of the frequencies.

2.2 Direct Coupling Analysis

Inverse statistical physics is based on the assumption that our MSA of natural sequences can be considered as a sample of a Boltzmann distribution.
Remembering that the MSA is a matrix $A = \{a_i^\mu, i = 1, ..., L, \mu = 1, ..., M\}$ which contains M sequences aligned over L positions, the Boltzmann distribution assigns a probability to each sequence of length L according to:

$$P(a_1, \ldots, a_L) = \frac{1}{Z} e^{-H(a_1, \ldots, a_L)} \tag{2.1}$$

where H represents the Hamiltonian and no temperature is needed in this setting.
Inverse statistical physics [16][3] aims at inferring the Boltzmann distribution from which sample A was generated.
Of course the correct form of the Hamiltonian H is unknown and with no prior knowledge $q^L - 1$ a priori independent probabilities need to be inferred.
This is not feasible and Maximum-Entropy Principle can be used to get models with less parameters. Starting from the dataset A, it is possible to compute the one and two point frequencies and impose that their data-derived empirical values coincide with the thermodynamic averages computed with respect to the sought distribution.
It can be shown that the maximum entropy distribution $P_{MEP}(x)$ that matches empirical frequencies is a Boltzmann distribution with a Potts Hamiltonian with form :

$$H(a_1, a_2, \ldots, a_L) = -\sum_{i \in V} h_i(a_i) \; - \sum_{(1 \leq i \leq j \leq L} J_{ij}(a_i, a_j) \tag{2.2}$$

The proof of this will be provided later. For now let us give an interpretation of the parameters.
$h_i(a_i)$ are the local fields. An high field $h_i(a_i)$ (of base a_i in site i) means that, if we generate sequences from our inferred model, $f_i(a_i)$ will be big. These fields model site conservation.
$J_{ij}(a_i, a_j)$ are the pairwise couplings and they model co-evolution. An high coupling $J_{ij}(a_i, a_j)$ (of bases a_i,a_j in sites i,j) means that the pair (a_i, a_j) appears very often in site i and j respectively. However, the direct relation between the f_ij and the J_ij is not evident, due to the collective effects present in the interacting systems given by Eq. 2.2.

So in our generated sequences $f_{i,j}(a_i, a_j)$ will be high.
For an homologous family of length L there are qL parameters for the local fields $h_i(a_i)$: there is a vector of fields. On the other hand there are $q^2|E|$ parameters for the interactions $J_{ij}(a_i, a_j)$ since each one of them is a $q \times q$ matrix.
Direct Coupling Analysis [17][18] models are built learning a Potts model (fields and couplings) from the data : MEL imposes that $h_i(a_i)$ and $J_{ij}(a_i, a_j)$ are chosen such that the marginals of eq. 2.1 match the empirical values given by the frequencies. In this case the problem is equivalent to find the Potts model with highest likelihood :

$$\mathcal{L}(h, J|A) = \frac{1}{N} \sum_{\vec{a} \in A} \log P(\vec{a}|h, J) \tag{2.3}$$

It is possible to infer the numerical values the parameters (local fields and pairwise couplings) by maximum likelihood.
Starting from this expression for $\mathcal{L}(h, J|A)$ we can compute $[h^*, J^*]$ maximizing it for example via Gradient Ascent : while the likelihood 2.3 is convex in the parameters, calculating it or its gradient is complicated (the partition function Z needs to be calculated for arbitrary parameter values).
After optimisation we get the parameters:

$$[h^*, J^*] = argmax_{h,J}\{\mathcal{L}(h, J|\mathcal{D}_N)\} \tag{2.4}$$

A lot of different DCA methods were developed in the literature with the aim of maximising efficiently the likelihood: depending on the setting and on the specific aim there are a lot of tools to approach this optimisation problem.

2.3 Phylogenetic Bias and Regularization

Generative models are built with the underlying assumption that all the sequences in an homologous family are drawn independently from a probability distribution $P_0(a_1, a_2, \dots a_L)$.
Actually often the sequences in a family do not respect this assumption : they are related phylogenetically.
The collected data do not explore homogeneously the sequence space and they are collected in a biased way[19]. In an homologous family there are sequences performing the same function and they come from different organisms; however it is assumed that there is a common ancestor and they can be very similar. For example the hemoglobin of the mice is only a few amino acids different from the human one.
On top of this sequenced species are unevenly selected based on the interest they have in the field. So some species are more represented with respect to others. The similarity between sequences of a given family was actually used to construct the phylogenetic tree of these sequences [20] but in our case the $P(a_1, a_2, \dots a_L)$ inferred from the family will show sharp peak in the region

of those sequences : this is a Phylogenetic bias.
To overcome this problem the proposed strategy is to compute the frequencies $f_i(a_i)$ and $f_{ij}(a_i, a_j)$ in a different way: a weight is assigned to all the sequences in the dataset.
The weight w_k associated to the k^{th} sequence of the family is equal to $w_k = \frac{1}{n_k}$ where n_k is the number of sequences that share more than 80% common nucleotides with sequence k. By doing this reweighting, we avoid "double-counting" very similar sequences, and obtain a more even sampling.
For example if there are 4 other sequences with more than 80% nucleotides in common, a weight $\frac{1}{5}$ will be assigned to all 5 of them.
The frequencies are then computed as:

$$f_i(a) = \frac{1}{N_{eff}} \sum_{k=1}^{N} \omega_k \cdot \delta_{a_i^k, a} \tag{2.5}$$

$$f_{ij}(a, b) = \frac{1}{N_{eff}} \sum_{k=1}^{N} \omega_k \cdot \delta_{a_i^k, a} \delta_{a_j^k, b} \tag{2.6}$$

with $N_{eff} = \sum_{k=1}^{N} \omega_k$.
The fact that not all sequences have the same importance changes the effective size of the homologous family dataset. If all the weights are equal to one the usual definition of the frequencies is retrieved.
On top of this there is another problem. Typical natural sequences have a length of hundreds of sites. This means that a huge amount if parameters still need to be inferred. Furthermore available samples are limited ($M = 10^2 - 10^5$ for a typical homologous family). These reasons make regularization necessary to avoid overfitting.
To get a better understanding of the problem let us propose an example. Suppose that nucleotides x and y are rarely encountered on sites i and j respectively ($f_i(x) = f_j(y) = 0.01$). If they evolve independently the probability of finding both nucleotides in sites i and j respectively of a sequence is equal to $0.01 \cdot 0.01 = 10^{-4}$.
At best our effective datasets are composed of few thousands sequences and such x,y occurrence maybe is not present at all.
Not only this shows an apparent anti-correlation but also an infinitely negative coupling between the two sites and nucleotides will be present.
An often used procedure to limit under-sampling effects is pseudo-count. Starting from empirical one and two point frequencies $f_i(a), f_{ij}(a, b)$ we impose:

$$f_i(a)' = (1 - \alpha) f_i(a) + \frac{\alpha}{q}$$

$$f_{ij}(a, b)' = (1 - \alpha) f_{ij}(a, b) + \frac{\alpha}{q^2}$$

The introduction of a pseudo-count is equivalent, on average, to extend the MSA with a fraction $\alpha(1 - \alpha)$ of sequences sampled uniformly in every site.
In this way all the occurrences are represented and there are no undesired,

infinitely negative parameters.
More sophisticated regularization schemes can be applied. Depending on the problem one can be better than others.

2.4 Maximum Entropy Principle and Boltzmann Machine DCA

Starting from an observation data-set $\vec{X}$ one goal of information theory is to gain information on the unknown probability distribution $P(\vec{x})$ from which the observation data-set was sampled.
When this dataset is limited in size i.e. frequencies cannot be inferred simply as empirical frequencies, it is possible to apply the Maximum Entropy Principle (MEP) [21].
After deciding some features of the observation data-set (often mean values of the empirical distribution) the aim of MEP is to find a probability distribution $P_{MEP}(\vec{x})$ that reproduces the selected features while being the most unconstrained.
Suppose we have N observed data points in our dataset $\vec{x}$:

$$\vec{x_1}, \vec{x_2}, \ldots, \vec{x_N}$$

M observables $O_\alpha(\vec{x})$ are selected and their average on the observations are computed as

$$\widetilde{O_\alpha} = \frac{\sum_{k=1}^{N} O_\alpha(\vec{x_k})}{N} \quad for \quad \alpha = 1, 2, \ldots, M$$

The goal is to find the probability distribution $P_{MEP}(\vec{x})$ with maximal Shannon entropy $S(\vec{x}) = -\sum_{\vec{x}} P(\vec{x}) \log\big(P(\vec{x})\big)$ that respects:

$$< O_\alpha(\vec{x}) >_{P_{MEP}} = \widetilde{O_\alpha} \quad for \quad \alpha = 1, 2, \ldots, M$$

A matching between average on observations and ensemble mean value of the selected observables is then imposed.
To do so we can use Lagrange multiplier technique to account for the constraints :

$$P_{MEP}(\vec{x}) \propto argmax_{P(\vec{x})} \left\{ S(\vec{x}) - \sum_{\alpha=1}^{M} \lambda_\alpha \left(\sum_{\vec{x}} P(\vec{x}) O_\alpha(\vec{x}) - \widetilde{O_\alpha} \right) \right\}$$

The solution is easily retrieved:

$$P_{MEP}(\vec{x}) \;\propto\; exp\Big\{ \sum_{\alpha=1}^{M} \lambda_\alpha O_\alpha(\vec{x}) \Big\} \tag{2.7}$$

The values of the λ_α have to be tuned keeping into account the constraints.
Once $P_{MEP}(\vec{x})$ is obtained, it can be used to sample artificial sequences.
The Boltzmann Machine Learning DCA [13] (bmDCA for short) is one of the

more successful generative models for homologous protein families and it is based on [22]. It is a maximum entropy generative model used to generate artificial biomolecules.
It is able to generate fully functional artificial sequences and it was also applied to predict the effects of mutations.
MSAs can be seen as matrices $A = \{a_i^\mu, i = 1, ..., L, \mu = 1, ..., M\}$ which contain M sequences aligned over L positions.
Following the maximum entropy principle, the model should replicate the single site frequencies $f_i(a)$ and the two sites frequencies $f_{ij}(a, b)$ of the training natural data (observation data-set).
Frequencies are given by the reweighted equations since one has to account for the phylogenetic bias.
Here the selected observables to be matched are $\delta(a_i, a)$ and $\delta(a_i, a)\delta(a_j, b)$ respectively. $\delta(a_i, a)$ poses the question if there is nucleotide a in position i.
Writing explicitly the mean values we obtain:

$$< \delta(a_i, a) >_{P_{MEP}} = \sum_{a_1, a_2, \ldots, a_L} P_{MEP}(a_1, \ldots, a_L)\delta(a_i, a) = P^i_{MEP}(a)$$

$$< \delta(a_i, a)\delta(a_j, b) >_{P_{MEP}} = \sum_{a_1, a_2, \ldots, a_L} P_{MEP}(a_1, \ldots, a_L)\delta(a_i, a)\delta(a_j, b) = P^{ij}_{MEP}(a, b)$$

Implying

$$P^i_{MEP}(a) = < \delta(a_i, a) >_{P_{MEP}}$$

$$P^{ij}_{MEP}(a, b) = < \delta(a_i, a)\delta(a_j, b) >_{P_{MEP}}$$

So the goal here is to infer the probability distribution with maximum entropy and with two point marginals that respect

$$P^i_{MEP}(a_i) = f_i(a_i)$$

$$P^{ij}_{MEP}(a_i, a_j) = f_{ij}(a_i, a_j)$$

The form of P_{MEP}, according to eq. 2.7, is:

$$P_{MEP}(a_1, a_2, \ldots, a_l) \propto exp\Big\{ \sum_i^L \sum_{a=1}^q h_i(a)\delta(a_i, a) + \sum_{i<j} \sum_{a=1}^q \sum_{b=1}^q J_{ij}(a, b)\delta(a_i, a)\delta(a_j, b) \Big\}$$

It is enough to exploit the properties of the delta function and the summation over a and b to rewrite it in a more compact form:

$$P_{MEP}(a_1, \ldots, a_l) = \frac{1}{Z} exp\Big\{ \sum_{i=1}^L h_i(a_i) + \sum_{i<j} J_{ij}(a_i, a_j) \Big\} \tag{2.8}$$

$$Z = \sum_{a_1, a_2, \ldots, a_L} exp\Big\{ \sum_{i=1}^L h_i(a_i) + \sum_{i<j} J_{ij}(a_i, a_j) \Big\} \tag{2.9}$$

It becomes evident that bmDCA is equivalent to a Potts model. Specifi-

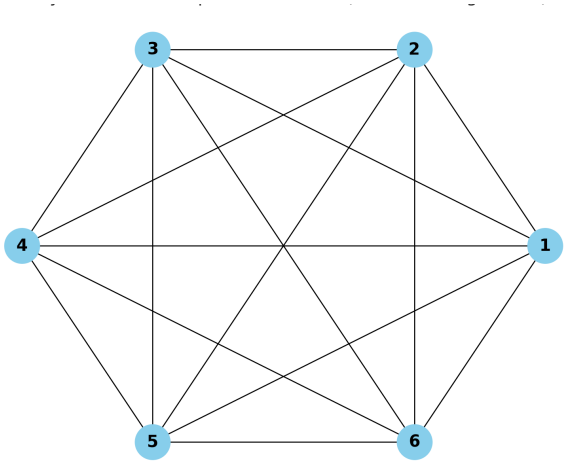

FIGURE 2.4: Fully connected model with 6 sites

cally a fully connected Potts model with Hamiltonian:

$$H(a_1,\dots,a_L) = -\sum_{i=1}^{L} h_i(a_i) - \sum_{i<j} J_{ij}(a_i,a_j) \tag{2.10}$$

$$P_{MEP}(a_1,\dots,a_L) = \frac{1}{Z} e^{-H(a_1,\dots,a_L)}$$

Figure 2.4 represents a fully-connected Potts model with 6 sites.
The computation of P_{MEP} marginals implies a sum over all the sequence space since the computation of the partition function is needed. This problem is known to be computationally hard and unfeasible for biomolecules of realistic length. So in order to find the best set of parameters $\{h_i, J_{ij}\}$ respecting moment matching conditions it is possible to apply a numerical approximated method.
For starters all the parameters are set to random values $J^0_{ij}(a_i,a_j)$ $h^0_i(a_i)$ and the following iterative procedure is adopted :

$$J^{t+1}_{ij}(a_i,a_j) = J^t_{ij}(a_i,a_j) + \eta\Big\{f_{ij}(a_i,a_j) - P^t_{ij}(a_i,a_j)\Big\}$$

$$h^{t+1}_{i}(a_i) = h^t_i(a_i) + \eta\Big\{f_i(a_i) - P^t_i(a_i)\Big\}$$

At each step t there is a probability distribution $P^t(a_1,\dots,a_L)$ with parameters $\{J^t_{ij}(a_i,a_j), h^t_i(a_i)\}$. Its first two marginals are $P^t_{ij}(a_i,a_j)$ $P^t_i(a_i)$ and a new iteration can be started: this procedure is a way to do likelihood maximization [13] and on the Maximum likelihood point the empirical averages match the ensemble ones. Now the higher is a parameter the higher is its associated marginal, so at each step it is enough to increase the parameters associated with marginals that are smaller than empirical frequencies and decrease the one which are higher.
The steady state of the algorithm is reached when $f_{ij} = p_{ij}$ (this also corresponds to Maximum Likelihood point).

The computations of the marginals is still an hard problem : the solution is to use MCMC (Monte Carlo Markov Chain) methods to sample sequences from $P^t(a_1, a_2, \ldots, a_L)$ and compute on them the one and two point frequencies. If the equilibrium condition is reached, they are a fair approximation of P^t_{ij} and $P^t_i(a_i)$.
Later MCMC methods will be discussed more thoroughly because they play a fundamental role in generative models.

2.5 Edge Activation DCA

bmDCA produces a fully connected model that tries to account for co-evolution between all possible pairs of residues, even when no actual co-evolution is occurring. This can lead to highly noisy $J_{ij}(a, b)$ in the coupling network and these are an artifact of the model.
To avoid this, in literature a parsimonious method called Edge Activation DCA (eaDCA) has been developed [12].
The goal is to build a sparse model activating couplings only between pairs which are really co-evolving. All other pairs should not be included to avoid noise overfitting .
This algorithm can be applied to a MSA A consisting of M sequences and it starts from an empty coupling network.
The resulting network $\mathcal{E}$, defined via the set of all non-zero couplings J_ij, is built iteratively by adding edges one by one. At each step a gradually more complex model is built until a statistical performance comparable to that of bmDCA is obtained.
In this framework likelihood maximization can be performed analytically and the entropy can be computed exactly.
Starting from a model where there are no edges ($\mathcal{E}_0 = \emptyset$), a series of edge sets $\mathcal{E}_t$ are constructed at each step by activating a new edge or updating an existent one. A pictorial representation of this is shown in Figure 2.5.
The model at step t will be :

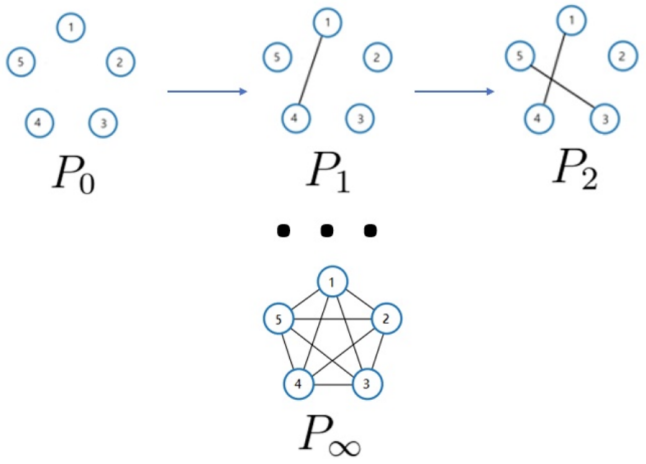

FIGURE 2.5: Edge set iterative construction : the limit is a full bmDCA. Figure courtesy of F. Calvanese.

$$P^t(\vec{a}) = \frac{1}{Z_t} e^{-E^t(\vec{a})}, \quad E^t(\vec{a}) = -\sum_{i=1}^{L} h_i(a_i) - \sum_{(i,j)\in\mathcal{E}_t} J_{ij}(a_i, a_j) \tag{2.11}$$

where E_t is called "statistical energy". The Likelihood of the model given the dataset $\mathcal{D}$ instead will be:

$$\mathcal{L}_t = \sum_{p=1}^{M} \omega_p \log P_t(a_1^p, \ldots, a_M^p) \tag{2.12}$$

where ω_k are training weights.
At t = 0 the distribution is given by:

$$P^0(\vec{a}) = \frac{1}{Z_t} \exp\{-\sum_{i=1}^{L} h_i(a_i)\}$$

while the log-likelihood $\mathcal{L}_0$ is maximized by choosing

$$h_i(a) = \log(f_i(a))$$

This simple model without couplings is known as profile model and the resulting partition function is $Z_0 = 1$.
At generic step t, there is the modification of a $q \times q$ coupling matrix J_{kl} on a single position pair (k, l). This results in a change in the statistical energy :

$$E^{t+1}(\vec{a}) = E^t(\vec{a}) - \Delta J_{kl}^*(a_k, a_l) \tag{2.13}$$

If (k, l) was not present in $\mathcal{E}_t$ there is an actual edge activation, otherwise it is only an edge update. The edge (k, l) and the coupling change $\Delta J_{kl}^*(a_k, a_l)$ are chosen to maximize the log-likelihood $\mathcal{L}_{t+1}$.
It can be proved that this is realized by choosing the pair

$$(k^*, l^*) \in argmax_{k,l} D_{KL}(f_{k,l} || P_{k,l}^t)$$

This is the the site pair for which the second moment P_{mn}^t differs the most from the empirical distribution f_{mn}.
Here D_{KL} denotes the standard Kullback-Leibler divergence :

$$D_{KL}(f||P) = \sum_{a,b} f(a,b) \log \frac{f(a,b)}{P(a,b)} \tag{2.14}$$

for any pair of probability distributions f and P.
To prove this we start defining $M_{eff} = \sum_k \omega_k$ and substituting eq. 2.11 into eq. 2.12 the expression for the likelihood becomes :

$$\log \mathcal{L}_t = -M_{eff} \log Z_t + \sum_{p=1}^{M} \omega_p (\sum_{i=1}^{L} h_i(a_i^p) + \sum_{i,j \in \mathcal{E}_t} J_{ij}(a_i^p, a_j^p))$$

Then, to go to step t+1 a modification $\Delta J_{kl}(a_k, a_l)$ is added and $\log \mathcal{L}_{t+1}$ reads:

$$\log \mathcal{L}_{t+1} = -M_{eff} \log Z_{t+1} + \sum_{p=1}^{M} \omega_p (\sum_{i=1}^{L} h_i(a_i^p) + \sum_{i,j \in \mathcal{E}_t} J_{ij}(a_i^p, a_j^p) + \Delta J_{kl}(a_k^p, a_l^p))$$

The goal is to pick the $\Delta J_{kl}(a_k, a_l))$ that maximises the likelihood gain

$$\Delta \log \mathcal{L} = \log \mathcal{L}_{t+1} - \log \mathcal{L}_t$$

$$\Delta \log \mathcal{L} = -M_{eff} \log \frac{Z_{t+1}}{Z_t} + \sum_{p=1}^{M} \omega_p \cdot \Delta J_{kl}(a_k^p, a_l^p)$$

Now it is possible to simplify this expression using delta functions and their properties:

$$\frac{\Delta \log \mathcal{L}}{M_{eff}} = -\log \frac{Z_{t+1}}{Z_t} + \sum_{p=1}^{M} \sum_{a,b} \omega_p \cdot \Delta J_{kl}(a_k^p, a_l^p) \delta_{a,a_k^p} \delta_{b,a_l^p}$$

Now remembering that $f_{kl}(a,b) = \frac{1}{M_{eff}} \sum_{p=1}^{M} \omega_p \cdot \delta_{a,a_k^p} \delta_{b,a_l^p}$ is the two-point frequency:

$$\frac{\Delta \log \mathcal{L}}{M_{eff}} = -\log \frac{Z_{t+1}}{Z_t} + \sum_{a,b} f_{kl}(a,b) \Delta J_{kl}(a,b)$$

Now the partition function terms can be computed via an ensemble average with respect to P_t. In fact

$$\frac{Z_{t+1}}{Z_t} = \frac{\sum_{\vec{a}} e^{-E_t(\vec{a})} \cdot e^{\Delta J_{kl}(a,b)}}{\sum_{\vec{a}} e^{-E_t(\vec{a})}} = \left\langle e^{\Delta J_{kl}(a_k, a_l)} \right\rangle_{P_t}$$

So

$$\frac{Z_{t+1}}{Z_t} = \sum_{a,b} e^{\Delta J_{kl}(a,b)} P_{kl}^t(a,b)$$

Then coming back to the likelihood gain :

$$\frac{\Delta \log \mathcal{L}}{M_{eff}} = -\log \sum_{a,b} e^{\Delta J_{kl}(a,b)} P_{kl}^t(a,b) + \sum_{a,b} f_{kl}(a,b) \Delta J_{kl}(a,b)$$

The final goal now is to maximise this expression with respect to $\Delta J_{kl}(a,b)$. To do so it is enough to perform a simple derivative and the result is :

$$-\frac{e^{\Delta J_{kl}^*(a,b)} P_{kl}^t(a,b)}{\sum_{c,d} e^{\Delta J_{kl}^*(c,d)} P_{kl}^t(c,d)} + f_{kl}(c,d) = 0$$

So that

$$\Delta J_{kl}^*(a,b) = \log \frac{f_{kl}(a,b)}{P_{kl}^t(a,b)} \tag{2.15}$$

Substituting this in the likelihood gain, the biggest possible gain is :

$$\frac{\Delta \log \mathcal{L}}{M_{eff}} = \sum_{a,b} f_{kl}(a,b) \log \frac{f_{kl}(a,b)}{P_{kl}^t(a,b)} = D_{KL}(f_{kl} || P_{kl}^t)$$

In the end

$$(k^*, l^*) = argmax_{k,l} D_{KL}(f_{kl}||P^t_{kl}) \tag{2.16}$$

As in standard bmDCA, computing the marginal distributions P^t_{kl} is a hard problem since it is equivalent to sum over all q^L possible sequences of aligned length L.
Markov Chain Monte Carlo (MCMC) sampling methods are needed to compute approximately the marginals.
$f_{ij}(a,b)$ can be zero because a couple of nucleotides (a,b) is never observed in sites i,j. If the edge (i,j) is targeted by eaDCA, this would lead to $J^*_{ij}(a,b) = -\infty$: in paractical applications, a few modifications have to be done.
Furthermore it could be that $P^t_{ij}(a,b) = 0$ because, during MCMC sampling, the couple (a,b) in sites i,j was never sampled. This yields $J^*_{ij}(a,b) = +\infty$.It is clear that regularization is needed.
Using pseudocounts as before, a regularized update is defined as :

$$\Delta J^*_{kl}(a,b) = \log \frac{(1-\alpha) f_{kl}(a,b) + \frac{\alpha}{q^2}}{(1-\alpha) P^t_{kl}(a,b) + \frac{\alpha}{q^2}} \tag{2.17}$$

The eaDCA algorithm needs a termination condition : the steady state is the same as bmDCA and it is reached when $p_{ij} = f_{ij}$, $\forall (i,j)$.
The empirical two-site covariances are :

$$c_{ij}(a,b) = f_{ij}(a,b) - f_i(a) f_j(b)$$

Instead the correlations of the model are given by :

$$c^t_{ij}(a,b) = P^t_{ij}(a,b) - P^t_i(a) P^t_j(b)$$

When there is a Pearson Correlation of 0.95 between $c_{ij}(a,b)$ and $c^t_{ij}(a,b)$ computed over all pairs of positions and nucleotides including the pairs $(i,j) \notin \mathcal{E}$, the algorithm stops. The choice of the $c^t_{ij}(a,b)$ instead of $f^t_{ij}(a,b)$ is dictated by the fact that generated sequences should replicate the statistics of the natural data. The one-point frequencies are easily checked. To test the two-point frequencies it is better to use the $c^t_{ij}(a,b)$ since this quantity isolates the co-evolutionary information contained in the $c^t_{ij}(a,b)$.
In fact it is possible to write $f_{ij}(a_i, a_j)$ as:

$$f_{ij}(a_i, a_j) = f_i(a_i) f_j(a_j) + \epsilon_{ij}(a_i, a_j) \tag{2.18}$$

where :

- $\epsilon = 0$ if the two sites are independent (do not co-evolve)
- $\epsilon > 0$ if the pair (a_i,a_j) is favoured by co-evolution
- $\epsilon < 0$ if the pair (a_i,a_j) is disadvantaged by co-evoluton

All the co-evolutionary information is contained in ϵ and

$$\epsilon_{ij}(a_i, a_j) = f_{ij}(a_i, a_j) - f_i(a_i) f j(a_j) \tag{2.19}$$

i.e. the connected correlation (or covariance) isolates conservation signals from co-evolution :

$$\epsilon_{ij}(a_i, a_j) = C_{ij}(a_i, a_j) \tag{2.20}$$

So using the correlation as a test for the correct reproduction of co-evolution statistics is a correct procedure. eaDCA preserves the partition function Z. Substituting the coupling update in the expression, we get :

$$\frac{Z_{t+1}}{Z_t} = \sum_{a,b} \exp\{\log \frac{f_{kl}(a,b)}{P^t_{kl}(a,b)}\} P^t_{kl}(a,b) \frac{Z_{t+1}}{Z_t} = \sum_{a,b} \frac{f_{kl}(a,b)}{P^t_{kl}(a,b)} P^t_{kl}(a,b) = 1$$

This proves that $Z_{t+1} = Z_t$. Now, since the starting point was a profile model with $Z_0 = 1$, at each time the model is normalized i.e. we have directly $P(a_1, \ldots, a_L) = \exp(-H(_1, \ldots, a_L))$ and this can be very practical when comparing sequences in different models, as done for example in homology detection in computational biology.

2.6 Gibbs Sampling and Importance Sampling

In the previous parts Monte Carlo Markov Chain methods[23] were often mentioned. Since they will play a significant role in the following, it is important to talk a bit about them. MCMC methods are needed: the computation of marginals of Boltzmann Distributions can be really slow. In fact the computation of the partition function is required and this means performing an exponential sum.
The evaluation of marginals is then performed thanks to MCMC methods : a big equilibrium sample is generated and from there estimating the marginal is far more efficient. There are many MCMC methods: here an introduction about the Gibbs Sampling algorithm is provided and then the focus will be on Importance Sampling.
The functioning of Gibbs Sampling is simple. We define the probability of a single nucleotide mutation, conditional to all other nucleotides as $P(a_i|\vec{a}_{-i})$ where a_i is the nucleotide in position i and $\vec{a}_{-i}$ is the sequence with nucleotide a_i removed:

$$P(a_i|\vec{a}_{-i}) = \frac{P(\vec{a})}{P(\vec{a}_{-i})}$$

the advantage of this distribution is that it is easy to normalise since it depends only on one random variable, namely a_i.
In our specific case the Boltzmann probability distribution $P(a_1, \ldots, a_L) = P(\vec{a})$ is known and in MCMC simulations we initialise random sequences of proteins.
The nucleotides in each site of each sequence are modified: the new nucleotide in position i is sampled from its conditional probability distribution

$P(a_i|A_{-i})$.
Once all the positions are sampled one time, a complete Gibbs sweep has been done.
The Gibbs algorithm satisfies the detailed balance condition, ensuring that the resulting Markov Chain has the target distribution as its equilibrium distribution.
Anyway, in these kind of methods reaching equilibrium is a delicate point, particularly in the sampling parts of training our models.
In the eaDCA case the likelihood is maximised iteratively and at each step a MCMC method is needed to sample some sequences. They are then used to compute a score between them and the training data to assess how good is the currently trained model.
If at each training step we start from random sequences and each time we wait for reaching equilibrium, the algorithm becomes really slow.
Fortunately the probability distribution changes only slightly from one training step to another (one edge maximum). Exploiting this, the "persistent contrastive divergence" strategy is adopted. It consists in using the previously sampled sequences as a starting point for the sampling of the next step. In this way only a few Gibbs sweeps are needed and the algorithm is way faster.
After the training, to assess the quality of our model, we need to sample from it starting from random sequences and to see if the score of the sequences is indeed comparable to the one obtained in the training. This process is called resampling.
In this setting the equilibrium is reached when the chains auto-correlation match the correlation between two independent chains (always using the same model).
The sampling method described generates an independent and identically distributed (iid) sample of sequences from the probability distribution.
As described in the introduction, in the Azoarcus setting we are interested in mutations from the wild-type that can be generated in laboratory. The problem of sampling model-informed mutations from a wild-type needs to be tackled.
First of all a concept of distance needs to be defined. In the following we will use the Hamming distance.
Given two sequences X and Y of same length N, the Hamming Distance is defined as

$$D_H(X,Y) = N - \sum_{i=1}^{N} \delta(X_i, Y_i) \tag{2.21}$$

where $\delta(x,y)$ is the usual Kronecker delta. It measures how many nucleotides are different between the two sequences.
The Importance Sampling algorithm provides a relatable way to generate mutations at each distance from a wild-type while keeping into account the model from which we are sampling. The aim is to generate mutations at distance K : it is needed to sample from $P(a_1, \ldots, a_L|K)$.
A naive idea would be to sample numerous sequences from the unconstrained $P(a_L, \ldots, a_L)$ and then filter those at distance k.

Unfortunately it could be that the probability of sampling a sequence at distance K is extremely low and an astronomically large equilibrium sample would be needed.
To overcome this problem a bias term is introduced in order to guide the probability distribution towards specific values of K.
Starting from a DCA model

$$P(a_1, \ldots, a_L) = \frac{1}{Z} \exp\{\sum_{i=1}^{L} h_i(a_i) + \sum_{i<j}^{L} J_{ij}(a_i, a_j)\} \tag{2.22}$$

The biased one reads:

$$P_\theta(a_1, \ldots, a_L) = \frac{1}{Z \cdot Z_\theta} \exp\{\sum_{i=1}^{L} h_i(a_i) + \sum_{i<j}^{L} J_{ij}(a_i, a_j) - \theta \cdot D_H(\vec{a}, \vec{x})\} \tag{2.23}$$

where Z_θ is the variation in terms of partition function, $D_H(\vec{a}, \vec{x})$ is the hamming distance between the sequence $\vec{a}$ and the wild-type $\vec{x}$ and θ is the bias strength.
The probability of sampling sequences close to the wild-type is higher when θ is positive and it is lower when θ is negative, pushing the sampling away from the wild-type.
By changing the values of θ and by doing different batches of sampling, it is possible to sample sequences at any desired distance K.
Eq. 2.23 can be written in a Potts way by including the bias term in the field term. In fact it is possible to define:

$$\bar{h_i}(a_i) = h_i(a_i) - \theta \cdot (1 - \delta_{a_i, x_i}) \tag{2.24}$$

In this way it can be shown that $P_\theta(a_1, \ldots, a_L)$ is again a Potts model :

$$P_\theta(a_1, \ldots, a_L) = \frac{1}{Z \cdot Z_\theta} \exp\{\sum_{i=1}^{L} \bar{h_i}(a_i) + \sum_{i<j}^{L} J_{ij}(a_i, a_j)\} \tag{2.25}$$

The bias term should guide the sampling without changing the original distribution. To address this, it can be proved that $P_\theta(a_1, \ldots, a_L|K) = P(a_1, \ldots, a_L|K)$.

2.7 ViennaRNA Package

The ViennaRNA Package [24] is an useful tool used in computational biology to study RNA. Inside there are a lot of different programs for a lot different applications. Since it will be helpful in the following, let us talk about RNA Secondary Structure Prediction and RNA Design.

The program for RNA Secondary Structure Prediction is called RNAFold

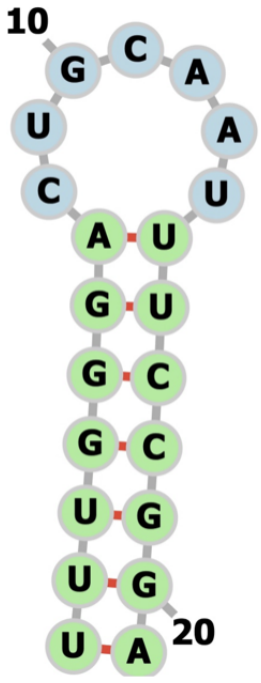

FIGURE 2.6: ViennaRNA package picture of a RNA hairpin with given dot-bracket structure.

and it predicts the minimum free energy (MFE) secondary structure for a given RNA sequence; it is the structure that, according to thermodynamic principles, has the lowest thermodynamic free energy. To predict it, Zuker dynamic-programming algorithm [25] is used.

The output of a MFE prediction is commonly represented in dot-bracket representation. In this notation, the RNA sequence is pictured with dots for unpaired bases and parentheses for paired bases.

On top of the secondary structure prediction, base pairing probabilities and the MFE value are provided thanks to the function RNAEval and these information will be very useful later.

ViennaRNA Package can also do the inverse: thanks to the function RNAInverse, taking as input a secondary structure in dot-bracket representation, it can generate sequences that fold in the desired structure using again thermodynamic principles.

Furthermore, by giving as inputs a sequence and a structure, it can compute the probability of the sequence folding into the structure and also the corresponding thermodynamic free energy can be computed.
We can define the thermoscore of a sequence as minus the thermodynamic free energy. In this way a low free energy will correspond to an high thermoscore : this is expected since good sequences have, by definition, low free energy.
Finally also graphical tools are present, capable of generating representations as the ones showed in Figure 2.6.

Chapter 3

Reintegration Methods

3.1 Reintegration of Heterogeneous Sequences

Generative statistical models have been presented as tools to generate artificial bio-molecular sequences : starting form an alignment of an homologous family of RNA or Proteins, Direct Coupling Analysis (DCA) models are built learning a Potts model via Likelihood maximisation.
High-throughput experiments became available and made possible to mass test biological sequences : three-dimensional structures, activities or thermodynamic stability can be measured in experiments and this allows to label sequences as functioning or not, making them "annotated".
An approach to reintegration of experimental data has been proposed before [26].
The goal was to integrate heterogeneous data in the Inverse Ising Problem. Although the setting is a little different, an Ising model can be seen as a Potts model with q=2.
Two different datasets were considered. The first dataset D_{eq} is a sample of spin configurations generated from an unknown "true" model H_0. In the biological setting these spin configurations correspond to the natural sequences on which DCA models are learnt on: it is enough to consider these as proteins with only two amino acids.
In general it is possible to suppose that natural sequences are generated from a "ground truth" model to which we do not have access; indeed the goal of statistical inference is to get closer to it.
The second dataset D_E is a collection of heterogeneous spin configurations, each one with a noisy measurement of its energy computed with respect to model H_0.
Then the energy measurement here is related to the quantity measured through the biological experiment in our setting: more on this will be discussed later.
$D_{eq} = \{s^1, ..., s^M\}$ dataset is composed of M configurations drawn at equilibrium from "true" model H_0 and it has empirical frequencies :

$$f_i^*(a) = \frac{1}{M}\sum_{\mu=1}^{M} s_i^\mu \quad , \quad f_{ij}^*(a,b) = \frac{1}{M}\sum_{\mu=1}^{M} s_i^\mu s_j^\mu \tag{3.1}$$

$D_E = \{\sigma^1, ..., \sigma^P\}$ dataset, instead, is composed by P arbitrary configuration

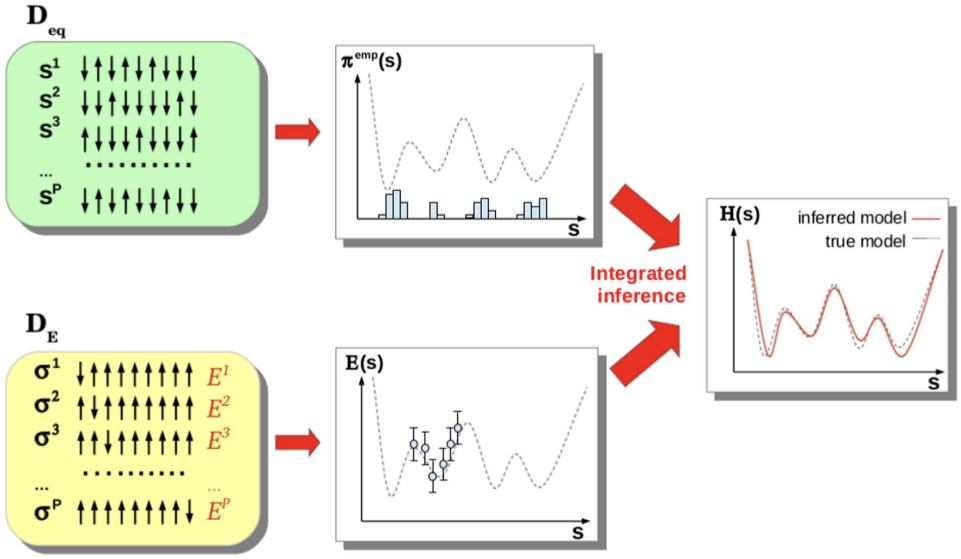

FIGURE 3.1: Representation of the inference setting datasets D_{eq} and D_E. The dashed black lines represent the actual unknown landscape while the red line the inferred landscape.Figure reproduced from [26]

and with each one there is also a noisy measurement of the energy.

$$E^a = H^0(\sigma^a) + \xi^a \tag{3.2}$$

Experimentally it is not possible to compute directly the energy of the sequences : to obtain it there must be some kind of mapping between the quantity experimentally measured and the Ising energy. This mapping surely is not perfect and ξ^a is a noise modelling uncertainties in the mapping and experimental noise. It is considered to be a white Gaussian noise with zero mean and variance the Δ^2.
A concrete example of mapping between experimental measures and energy will be provided later.
Using these dataset a new form of the Likelihood was devised to infer a model H hopefully closer to the true model H_0 :

$$\mathcal{L}(h, J|D_{eq}, D_E) = \log P(D_{eq}|h, J) + \log P(D_E|h, J) \tag{3.3}$$

The expression of $P(D_{eq}|h, J)$ is given by a Boltzmann Distribution :

$$P(D_{eq}|h, J) = \exp\left\{ -\sum_{\mu=1}^{M} H(s^\mu) - M \cdot \log Z(h, J) \right\} \tag{3.4}$$

Instead The expression $P(D_E|h, J)$ was obtained thanks to an integral over the Gaussian distribution of the noise ξ^a.

$$P(D_E|h, J) = \prod_{a=1}^{P} \int d\xi^a P(\xi^a)\delta(E^a - H^a - \xi^a) \tag{3.5}$$

$$P(D_E|h,J) = \frac{1}{(2\pi\Delta^2)^{\frac{p}{2}}} \exp\left\{ -\sum_{a=1}^{p} \frac{[E^a - H(\sigma^a)]^2}{2\Delta^2} \right\} \tag{3.6}$$

Now maximising the Likelihood a set of self consistent equations is found :

$$p_i(a) = f_i^*(a) + \frac{\lambda}{1-\lambda}\frac{1}{M}\sum_{a=1}^{P} \sigma_i^a [E^a - H(\sigma^a)] \tag{3.7}$$

$$p_{ij}(a,b) = f_{ij}^*(a,b) + \frac{\lambda}{1-\lambda}\frac{1}{M}\sum_{a=1}^{P} \sigma_i^a \sigma_j^a [E^a - H(\sigma^a)] \tag{3.8}$$

They are indeed self consistent because $p_i(a) = \langle \sigma_i \rangle_H$ and $p_{ij}(a,b) = \langle \sigma_i \sigma_j \rangle_H$ depend on H itself. This is not optimal and it can be a huge problem when trying to solve for $p_i(a), p_{ij}(a,b)$.
$\lambda = \frac{1}{1+\Delta^2}$ is an hyperparameter accounting for the strength of the interaction : if our measure is very noisy (big Δ^2) the sequence has a small effect in changing the moments. So the case $\lambda = 0$ (large noise) corresponds to the case in which no reintegration is done and the standard Inverse Ising is recollected. On the contrary the case $\lambda = 1$ corresponds to noiseless energy measurements.
The idea is that if our model assigns already the correct energy to a sequence in D_E then no correction to the empirical frequencies is needed. On the contrary if our model is wrong in the prediction of the energy then the moments are corrected proportionally to the difference in energy.
The take-home message is that it is possible to reintegrate data into our model. The new probability distribution is obtained by maximising another objective function and its first two moments are functions of the original empirical frequencies, adjusted with an experimental feedback.

3.2 Reintegration of Negative Sequences

To go forward in the direction of reintegrating annotated sequences let us start from a Potts model H_1 learnt via DCA on some dataset D_1 of sequences belonging to some homologous family.
Dataset D_1 has empirical frequencies f_i, f_{ij} and it can be thought as a small dataset sampled from a ground truth model H_0 which is unknown and it is actually the final goal of statistical inference.
In general dataset D_1 is not enough to infer H_0 but surely it is possible to learn a DCA model H_1 from it.
From this model we can generate via Gibbs Sampling a new dataset D_2 which will be strongly related to the sequences of D_1 and it will be regarded as test dataset. This setting is represented in Figure 3.2.
We now suppose to do a to an experiment on the sequences so that $\forall \vec{a} \in D_2$ a positive or negative label is assigned to it depending on its energy E_0 measured with respect to ground truth model H_0. This is shown in Figure 3.3.
Labelling sequences is equivalent to grasp how the ground truth model classifies the generated sequences: since H_0 is our (unknown) ground truth from

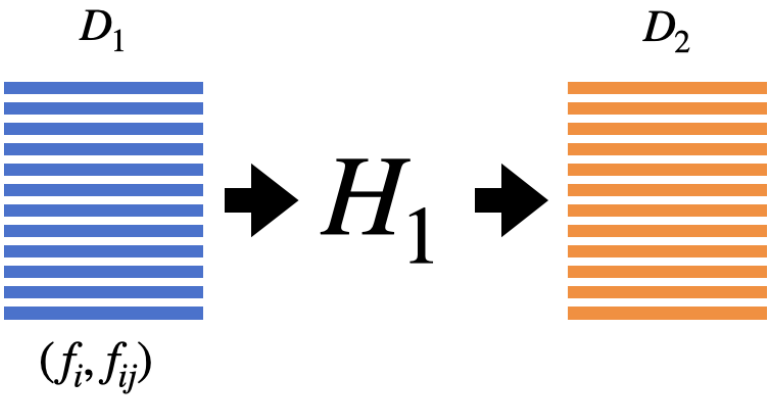

FIGURE 3.2: Representation of the setting : D_1 is the dataset on which H_1 is learnt on, D_2 is the dataset sampled from H_1

which dataset D_1 was generated (all are functioning in D_1), there would be some threshold E^* value saying which sequence is labeled negative or positive.
In the following the non functioning sequences (energy E_0 above the threshold E^*) will be called "negatives" while the functioning sequences (energy E_0 below the threshold E^*) will be referred as "positives". Computing the energy with respect to H_0 is like measuring a sort fitness of the generated sequences without any real experiment: it is a fitness proxy. The usage of fitness proxy will be crucial in the following.
To exploit this information to improve our model H_1 we can isolate the negative dataset made only of sequences with a negative label and compute their frequencies f_i^-, f_{ij}^- .
The sequences belonging to the negative dataset are the most informative ones: they are not random and our model H_1 deems them as good. Despite this they do not pass the experimental test because in reality our model was not able to grasp some fundamental details.
To use this information our dataset D_1 (in which each sequence has a training weight $w \leq 1$) can be extended with the sequences belonging to the negative dataset.
Similarly to what was done to get rid of the phylogenetic bias during the

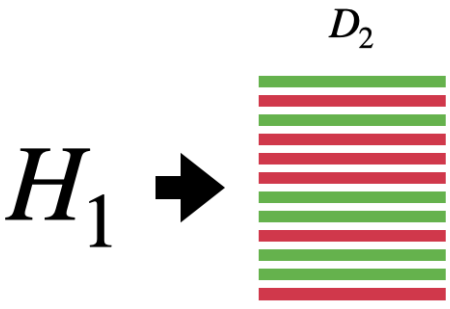

FIGURE 3.3: Representation of test dataset D_2 after labelling

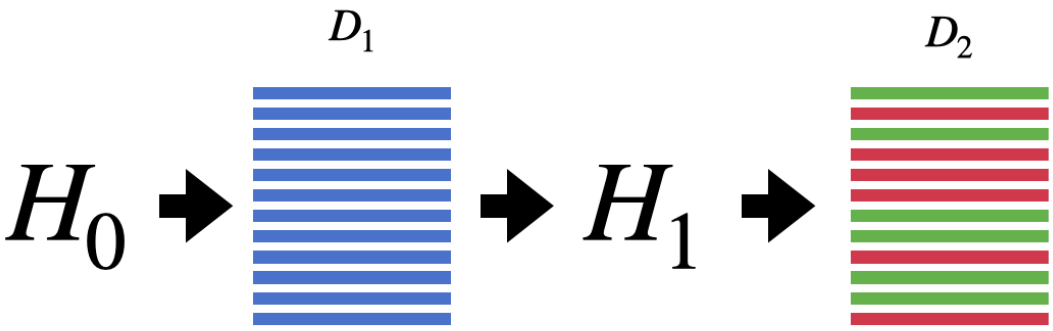

FIGURE 3.4: Full Setting : H_1 is our inferred model, H_0 is unknown.

training of DCA models, it is possible to do so by assigning training weight $w = -1$ to all of them. Actually it is wise to introduce an hyper-parameter λ representing the integration strength and set $w = -\lambda$.
In this way the augmented dataset will be more informative and it will have different frequencies from f_i, f_{ij}.
Keeping in mind the formulae for the re-weighted frequencies, it is trivial to see that the new frequencies for the augmented dataset are given by :

$$f_i^*(a) = \frac{f_i(a) - \lambda f_i^-(a)}{1 - \lambda} \tag{3.9}$$

$$f_{ij}^*(a,b) = \frac{f_{ij}(a,b) - \lambda f_{ij}^-(a,b)}{1 - \lambda} \tag{3.10}$$

The interpretation of these formula is the following : they are function of the original frequencies but they are adjusted according to experimental results : more precisely frequencies $f_i(a)$ and $f_{ij}(a,b)$ (leading to many non-functional sequences) get reduced into the adjusted frequencies 3.9 and 3.10.
At this point a new model can be trained using f_i^*, f_{ij}^* as target. Actually this case can be seen as a particular case of a more general setting so the training phase will be addressed later.

3.3 Objective Function Design

A bmDCA model $P^1(\vec{a})$ is trained maximising the Natural Data $\mathcal{D}_N$ log-likelihood $\mathcal{L}(\mathcal{D}_\mathcal{N})$

$$\mathcal{L}(h, J|\mathcal{D}_N) = \frac{1}{N} \sum_{\vec{a} \in \mathcal{D}_N} \log P^1(\vec{a}|h, J) \tag{3.11}$$

The set of parameters $\{h, J\}$ that maximizes $\mathcal{L}(h, J|\mathcal{D}_N)$ is found analytically or numerically. The corresponding Boltzmann distribution $P^1(\vec{a})$ is then known. Remembering the Maximum Entropy Principle we know that the first and second moment of $P^1(\vec{a})$ must match the empirical frequencies f_i, f_{ij} of the Natural Data $\mathcal{D}_N$.
This set of parameters can be seen as a generative model from which we can sample a dataset of artificial sequences $\mathcal{D}_T$ that can be tested experimentally: a fraction of them will be biologically functional while the others will not.
The question is how to use this information to improve the DCA model $P^1(\vec{a})$. To do so the strategy is to learn a new model $P^2(\vec{a}|h, J)$ using the following objective function to be maximised:

$$\mathcal{Q}(\mathcal{D}_N, \mathcal{D}_T, \lambda) = \frac{1}{N} \sum_{\vec{a} \in \mathcal{D}_N} \log P^2(\vec{a}|h, J) + \frac{\lambda}{M} \sum_{\vec{b} \in \mathcal{D}_T} \log P^2(\vec{b}|h, J) \cdot w(\vec{b}) \tag{3.12}$$

The first term is exactly the log-likelihood maximization on the Natural Data of eq. 3.11 : if there is no access to experimental data, meaning that $\mathcal{D}_T = \emptyset$, the objective function reverts back to the standard log-likelihood.
Instead the second term is the key to reintegration and it is acting on the probabilities of the tested sequences relying on the value of the adjustment function w : a value of w is assigned to every sequence $\vec{b} \in \mathcal{D}_T$. The intensity of the tuning is controlled by the hyperparameter λ so that if $\lambda = 0$ the classic inverse problem is recovered.
In general $w(\vec{b})$ is arbitrary but it has to follow the following rules:

- $w(\vec{b})$ should be negative for the sequences that our original model $P^1(\vec{a})$ deems as good but fail the experimental functionality test. In this way the maximization of the second term of the eq. 3.12 will try to reduce the probabilities $P^2(\vec{b})$ that the model assigns to them. Once again these are the more informative ones.
  In principle $w(\vec{b})$ should be negative also for sequences that our model deems as bad and that do not pass the experimental functionality test but it is safe to assume that the sequences generated from $P^1(\vec{a})$ are considered as good for the model itself.

- $w(\vec{b})$ should be positive for the sequences that our original model $P^1(\vec{a})$ deems as good and pass the experimental functionality test. In this way the maximization of the second term of the eq. 3.12 will try to increase the probabilities $P^2(\vec{b})$ that the model assigns to them.

Again $w(\vec{b})$ should be positive also for sequences that our model deems as bad but pass experimental functionality test.

This derivation is general and it does not depend on the specific form of $w(\vec{b})$. Anyway in the following numerous examples will be provided.
Going back to eq. 3.12 for the Objective Function and substituting the expression of the Boltzmann Equation 2.1:

$$\mathcal{Q} = -\frac{1}{N}\sum_{\vec{a}\in\mathcal{D}_N} H_2(\vec{a}) - \log Z_2 - \frac{\lambda}{M}\sum_{\vec{b}\in\mathcal{D}_T} H_2(\vec{b})\cdot w(\vec{b}) - \frac{\lambda}{M}\log Z_2 \sum_{\vec{b}\in\mathcal{D}_T} w(\vec{b})$$

The aim is to maximise the Objective Function so the partial derivatives with respect to the parameters have to be computed.
Similarly to eq. (cap2) the terms involving the derivatives of the Potts Hamiltonian lead to :

$$-\frac{1}{N}\sum_{\vec{a}\in\mathcal{D}_N}\frac{\partial H_2(\vec{a})}{\partial h_i(a)} = f_i(a)$$

$$-\frac{1}{N}\sum_{\vec{a}\in\mathcal{D}_N}\frac{\partial H_2(\vec{a})}{\partial J_{ij}(a,b)} = f_{ij}(a,b)$$

The derivatives of the partition function instead :

$$-\frac{\partial \log Z_2}{\partial h_i(a)} = P_i^2(a)$$

$$-\frac{\partial \log Z_2}{\partial J_{ij}(a,b)} = P_{ij}^2(a,b)$$

Finally the terms involving the derivatives with the adjustment function $w(\vec{b})$:

$$-\sum_{\vec{b}\in\mathcal{D}_T}\frac{\partial H_2(\vec{b})\cdot w(\vec{b})}{\partial h_i(a)} = \sum_{\vec{b}\in\mathcal{D}_T}\delta_{b_i,a}\cdot w(\vec{b})$$

$$-\sum_{\vec{b}\in\mathcal{D}_T}\frac{\partial H_2(\vec{b})\cdot w(\vec{b})}{\partial J_{ij(a,b)}} = \sum_{\vec{b}\in\mathcal{D}_T}\delta_{b_i,a}\cdot\delta_{b_j,b}\cdot w(\vec{b})$$

Rearranging terms the following equations for the first and second moment of $P^2(\vec{a}|h,J)$ are found:

$$P_i^2(a) = \frac{f_i(a) + \frac{\lambda}{M}\sum_{\vec{b}\in\mathcal{D}_T}\delta_{b_i,a}\cdot w(\vec{b})}{1 + \frac{\lambda}{M}\sum_{\vec{b}\in\mathcal{D}_T} w(\vec{b})} \tag{3.13}$$

$$P_{ij}^2(a,b) = \frac{f_{ij}(a,b) + \frac{\lambda}{M}\sum_{\vec{b}\in\mathcal{D}_T}\delta_{b_i,a}\cdot\delta_{b_j,b}\cdot w(\vec{b})}{1 + \frac{\lambda}{M}\sum_{\vec{b}\in\mathcal{D}_T} w(\vec{b})} \tag{3.14}$$

The new moments of $P^2(\vec{a}|h,J)$ are function of the moments of $P^1(\vec{a}|h,J)$ but they are adjusted depending on the experimental data.
To grasp what is happening it is useful to remember that the sequences generated from $P^1(\vec{a}|h,J)$ are not random but they are actually generated from a DCA model that includes site conservation and site co-evolution. The negative sequences are almost functional but there must be some details that our model $P^1(\vec{a}|h,J)$ failed to grasp. This is mostly due to the quality and the quantity of the data on which $P^1(\vec{a}|h,J)$ was trained on. Indeed homologous families are composed of relatively few data (order of thousands) and a strong phylogenetic bias is present.
This also justifies the need of these reintegration methods.
Here all sequences are reintegrated : the good ones with a positive adjustment and the negative ones with a negative adjustment.
In the parts where the model $P^1(\vec{a}|h,J)$ was correct there will not be many changes because the model was already grasping them with the empirical frequencies and both the negative and positive ones are showing the right bases. In the parts where the model $P^1(\vec{a}|h,J)$ was wrong the frequencies will change more because the good sequences will enforce the pre-existent signal while the negative ones will actively adjust the frequencies.
$P_i^2(a)$ and $P_{ij}^2(a,b)$ can be interpreted as new target frequencies and a new DCA model can be trained using the same training strategies used before i.e. bmDCA or eaDCA.
It is important to notice that this may be dangerous since $P_i^2(a)$ and $P_{ij}^2(a,b)$ may become negative or larger than 1 in some cases. This is mostly relevant in case of very strong reintegration, i.e. rather large λ. This is the strength of the proposed procedure since here, differently from the reintegration techniques present in the Deep-Learning framework, no back-propagation is needed and this helps both interpretability and algorithm efficiency; in fact it is possible to use the same training strategy used in standard DCA.
Then from $P_i^2(a)$ and $P_{ij}^2(a,b)$ a new DCA model H_2 can be trained and from this a new set of artificial sequences can be generated and hopefully it performs better experimentally.
This procedure is powerful and well-founded : before showing its application to both synthetic and real data, two example of adjustment function $w(\vec{b})$ can be provided and it is possible to recover both the settings described before : the reintegration of hetereogeneous and negative data.

3.4 Example : Negative Sequences

If $\mathcal{D}_T$ is composed only by sequences that did not pass the experimental test, it will have empirical frequencies f_i^-, f_{ij}^-.
It is possible to assign $w = -1 \quad \forall \vec{a} \in \mathcal{D}_T$. and in this case eqs 3.13 and 3.14 simplify a lot :

$$P_i^2(a) = \frac{f_i(a) - \lambda f_i^-(a)}{1-\lambda}$$

$$P_{ij}^2(a,b) = \frac{f_{ij}(a,b) - \lambda f_{ij}^-(a,b)}{1-\lambda}$$

These equations are equivalent to eqs. 3.9 and 3.10 and reintegrating negative sequences is then a particular case of eqs 3.13 and 3.14.

3.5 Example : Reintegration with Potts Energy

Starting again from a Potts model H_1 learnt via DCA on some dataset D_1 as we did in the reintegration of negative sequences, it is possible to provide another relevant example of adjustment function w.
Dataset D_1 has empirical frequencies f_i, f_{ij}, computed via eqs. 2.5 and 2.6.
From this model we can generate via Gibbs Sampling a new dataset D_2 which will be somehow related to the sequences of D_1 and will be regarded as test dataset.
Now thanks to biological experiments each tested sequence is provided with a measure of fitness or activity and also its energy E_1 with respect to model H_1 can be computed. A mapping from the fitness measures can be done to find the energies E_0 of these sequences computed with respect to a "ground truth" model H_0. In this way we can compare the energies from the model H_1 with those computed from H_0 and design the adjustment consequently. A good sequence will have high fitness/low energy. Sorting the fitness measures in decreasing order and the energies E_1 in increasing order, it is possible to assign the lowest energy to the sequence with highest fitness, substituting each fitness value with the value of the energy assigned to it. Then the same is done assigning the second lowest energy E_1 to the sequence with the second higher fitness and so on. In this way the energies E_0 will be an experimentally-informed shuffle of the energies E_1.
Based on this mapping an adjustment function can be designed : after assigning a new energy E_0 to each of the tested sequences with this mapping we can take $w = E_1 - E_0$ so that if a sequence had a low energy E_1 but also low fitness (i.e. high energy E_0) it will be largely penalized in the training and viceversa if a sequence had a high energy E_1 but also high fitness it will be promoted in the training. Substituting this in eqs. 3.13 and 3.14 the result is :

$$P_i^2(a) = \frac{f_i(a) + \frac{\lambda}{M}\sum_{\vec{b}\in\mathcal{D}_T}\delta_{b_i,a}\cdot\left(E_1(\vec{b}) - E_0(\vec{b})\right)}{1 + \frac{\lambda}{M}\sum_{\vec{b}\in\mathcal{D}_T}\left(E_1(\vec{b}) - E_0(\vec{b})\right)} \tag{3.15}$$

$$P_{ij}^2(a,b) = \frac{f_{ij}(a,b) + \frac{\lambda}{M}\sum_{\vec{b}\in\mathcal{D}_T}\delta_{b_i,a}\cdot\delta_{b_j,b}\cdot\left(E_1(\vec{b}) - E_0(\vec{b})\right)}{1 + \frac{\lambda}{M}\sum_{\vec{b}\in\mathcal{D}_T}\left(E_1(\vec{b}) - E_0(\vec{b})\right)} \tag{3.16}$$

These equations are strikingly similar to eqs. 3.7 and 3.8. So the reintegration of heterogeneous sequences can be seen as a particular case of our setting. Furthermore these new equations are not self consistent and the moments of the distribution can be easily computed.

3.6 Edge Activation DCA with new Objective Function

Similar to what was done in the eaDCA setting, it should be possible to find out what edge to activate in order to maximize the new objective function in eq. 3.12. By starting from the new expression for the two-point frequency

$$P^2_{ij}(a,b) = \frac{f_{ij}(a,b) + \frac{\lambda}{M}\sum_{\vec{b}\in\mathcal{D}_T}\delta_{b_i,a}\cdot\delta_{b_j,b}\cdot w(\vec{b})}{1+\frac{\lambda}{M}\sum_{\vec{b}\in\mathcal{D}_T} w(\vec{b})}$$

it is possible to perform the same computations with the Objective Function gain performed in Ch.2.
The results are, intuitively:

$$(k^*,l^*) \in argmax_{k,l} D_{KL}(P^2_{kl}||P^t_{kl}) \tag{3.17}$$

$$\Delta J^*_{kl}(a,b) = \log\left(\frac{P^2_{kl}(a,b)}{P^t_{kl}(a,b)}\right) \tag{3.18}$$

Chapter 4

Reintegration: Fitness proxies and synthetic data

Since biological experiments can take a lot of time and they can be very expensive, in the development of our method it is very important to have a good proxy for the experimentally measured quantities like fitness or activity measures. In the following two examples of fitness proxies are used : Potts energy and ViennaRNA thermo-score.
Thanks to these fitness proxies it is possible to apply and test the reintegration methods that were presented in Chapter 3.
The aim of this chapter is to see some results of reintegration of synthetic data tested with a fitness proxy: this not only proves the validity of our methods but it also encourages their application to real data with real experimental measures.
The utilization of synthetic data is very frequent in biology. Synthetic data are very useful when obtaining authentic data is challenging due to limited availability and they are a good starting point when approaching a new problem.

4.1 Using Potts Energy as Fitness Proxy

Given a Potts model $H = \{h, J\}$ it is possible to evaluate, with respect to the model itself, the energy of every sequence belonging to the huge sequence space.
Sequences with a low energy will be deemed as good by the model while sequences with an high energy will be regarded as bad .
With this principle in mind let us go back to the setting described in the reintegration methods.
The best datasets accessible nowadays are the MSAs of homologous families and it is possible to learn DCA models from them. A Potts model H_1 is learnt via DCA on some dataset D_1 of sequences belonging to some homologous family (natural data).
Dataset D_1 can be thought as a small dataset sampled from a ground truth model H_0 which is unknown and it is actually the final goal of statistical inference.
The general goal of our reintegration methods is to improve model H_1 learnt

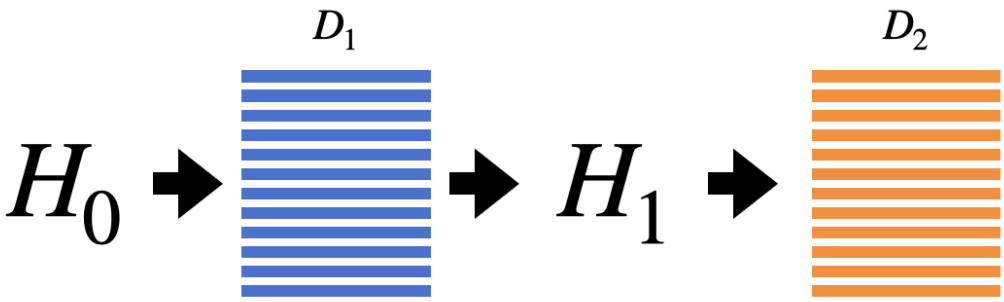

FIGURE 4.1: Setting : H_0 is the ground truth model from which the natural data D_1 are sampled, H_1 is the Potts model learnt on the natural data and D_2 are the artificial sequences sampled from H_1.

on natural sequences D_1 by reintegrating experimental tests based on some fitness measure. In this paragraph instead the aim is mostly testing our methods : since doing biological experiments is expensive and long we need a fitness proxy.
For sure the model H_1 is less informative than ground truth unknown model H_0 since it is only a Potts model matching the frequencies of D_1. Anyway, in the right setting, H_1 may take the role of ground truth model and it can be a good first proxy to test artificial sequences.
Let us describe a setting in which H_1 can assume the role of ground truth model. From model H_1 we can generate via Gibbs Sampling a new dataset D_2 which will be strongly related to the sequences of D_1. This is shown in figure 4.1.
Now the training of H_1 stopped when a 95 percent correlation was present between the correlation matrix C^2_{ij} of the sampled sequences D_2 and the C^1_{ij} of the D_1 natural dataset. Then, if a second Potts model H_2 is learnt on the whole D_2 dataset, it will be very similar to model H_1 and at that point H_1 would lose the role of ground truth model.
We can take a small fraction of D_2 dataset and learn a second Potts model H_2 from it. The size of the fraction of D_2 is problem-dependent but, during the analysis, it was observed that taking a tenth of the sequences of D_2 leads to good results and H_2 and H_1 are appreciably different: model H_2 will be worse than model H_1.
Now we want to apply our methods to improve H_2 so a dataset D_3 is sampled from it. In this scenario energies computed with respect to the model H_1 can be regarded as a fitness proxy to test sequences sampled from H_2.
Now it is possible to compute the energies E of sequences belonging to D_3 with respect to the assumed ground truth model H_1. Starting from H_2, our reintegration methods can be applied with the hope to get a better model H^* capable to generate sequences D^* with smaller average energy with respect to H_1 than those generated from H_2. This is shown in figure 4.2.
To get the new target frequencies on which H^* will be learnt it is enough to

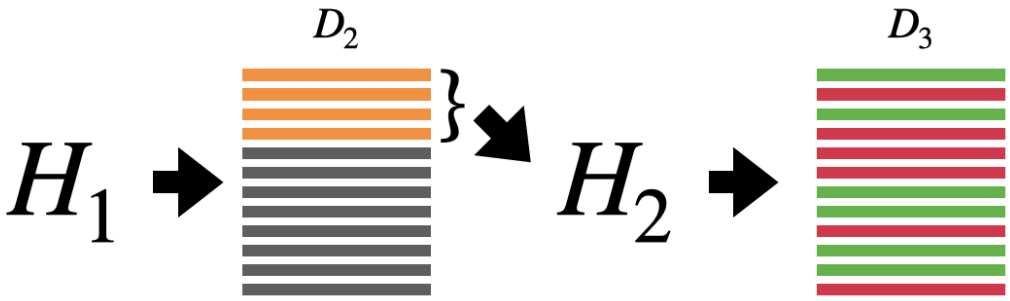

FIGURE 4.2: H_1 is now thought as the ground truth model and D_2 is sampled from it. H_2 is learnt on a fraction of D_2 and D_3 is the test dataset sampled from H_2.

apply eqs.3.13 and 3.14 using as Adjustment Function $w(\vec{b}) = -E \; \forall \vec{b}$ in D_3. Intuitively an high energy means that the sequence is bad so $w(\vec{b})$ needs to be negative to penalize the sequence during the training.

Using these new adjusted frequencies as target it is possible to train the

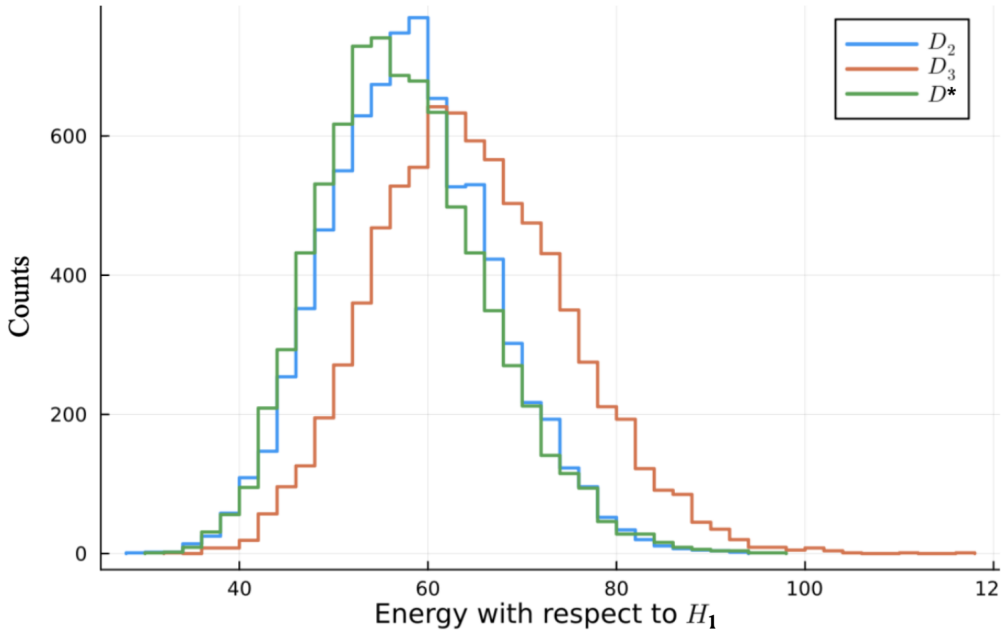

FIGURE 4.3: Histogram of the energies computed with respect to H_0 of the three artificial dataset related to the family of glycine riboswitches (RF00504).

sought model H^*. An optimization using different values of hyperparameter λ is needed but after doing this H^* is learnt via eaDCA and a dataset D^* is sampled from it.

This was done for several RNA families and with different sizes of the artificial datasets: the histogram shown in Figure 4.3 was done for the family of glycine riboswitches (RF00504) which consists of 4556 sequences.

The average energy with respect to model H_1 of sequences in D^* is far lower than the average energy of sequences in D_3 and it is comparable (depending on the specific setting) to that of the sequences in D_2 on which the original model H_2 was learnt on.
Then the new model H^* does perform better than H_2: the reintegration of data labelled thanks to model H_1 was successful.
After this more sophisticated fitness proxies needed to be investigated.

4.2 Reintegration with ViennaRNA thermo-score

ViennaRNA package [24] is very useful in our setting. Given a dot-bracket secondary structure, it can generate thousands of sequences that supposedly fold in that specific structure using RNAInverse function.
Not only this, ViennaRNA package is also able to take as input a sequence and a structure and output the probability of that sequence folding in that specific structure with RNAEval function.
Actually RNAEval can do more: given a structure, it can compute the thermodynamic free energy of a sequence. This, with a change of sign, is a very good fitness proxy since a lot of details are taken into account when computing this thermodynamic free energy. Minus the thermodynamic free energy is called the thermo-score so that the lower the free energy, the higher the thermo-score.

Starting from a dot-bracket structure, it is possible to generate a dataset

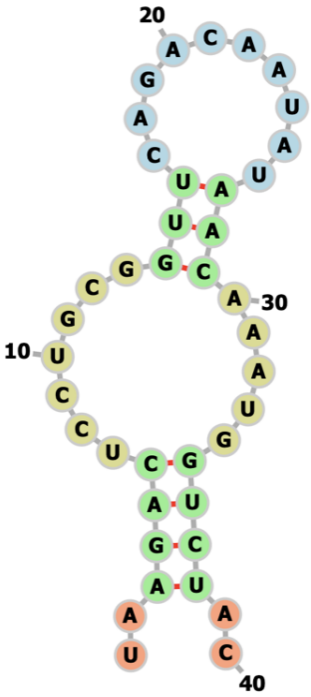

FIGURE 4.4: ViennaRNA package picture of an hammerhead ribozyme

D_V from ViennaRNA package with RNAInverse. From this we can learn an eaDCA model H_V as usual.
Now, from H_V, it is possible to generate artificial sequences D_A via Gibbs Sampling.
Now the procedure is simple : for each sequence in D_A we can evaluate its thermo-score and use this as adjustment function for our reintegration methods. In this way we can compute the new target frequencies and learn a second model H_R, product of the reintegration.
From H_R, a new dataset D_R can be sampled and tested : the sequences belonging to D_R are expected to have an higher thermo-score with respect to those of D_A.
This pipeline was applied to a 40 base pairs long molecule, the hammerhead ribozyme (Figure 4.4), and the results are shown in figure 4.5 : not only the sequences from D_R perform better than the sequences in D_A but, in terms of thermo-score, they outperform even those generated by ViennaRNA package.
Again, we conclude that the reintegration was successful and our model even outperforms RNAInverse. With this second test, which uses a ground truth not represented by a Potts model, we gained sufficient confidence to apply the reintegration method to real data and to generate new sequences for experimental testing, cf. Section 5.

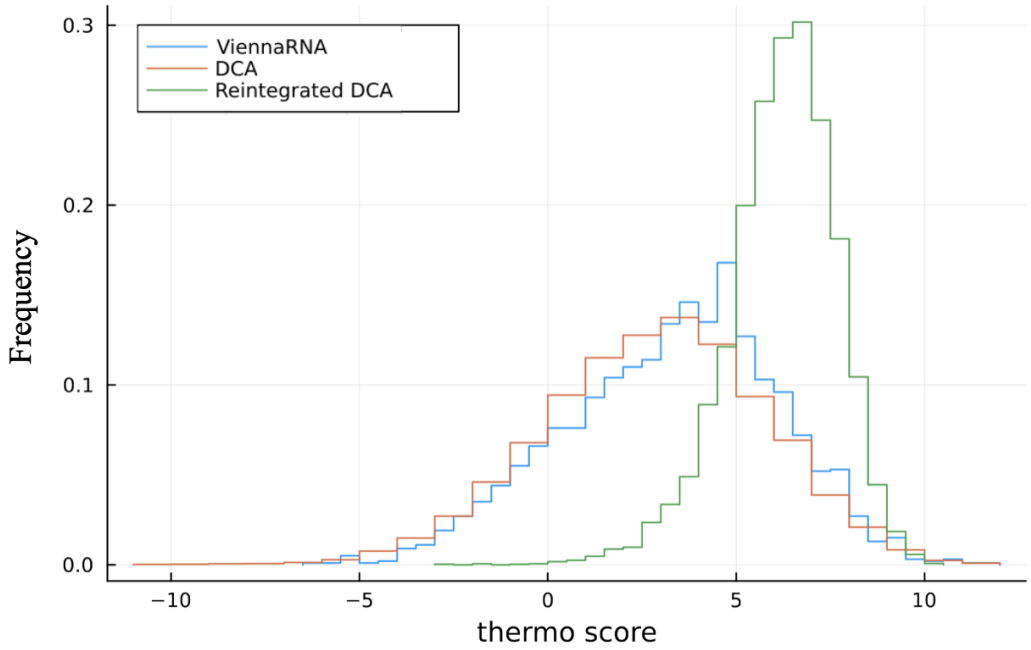

FIGURE 4.5: Histogram of the ViennaRNA thermo scores computed for three dataset: the sequences generated from ViennaRNA package, the sequences generated from DCA and the sequences generated after integration.

4.3 Diversity and Entropy loss

One of the strengths of eaDCA[12] is the possibility of computing the entropy of a model $P(a_1, \ldots, a_L)$.
Thanks to the entropy it is possible to know how many functional RNA sequences model can generate. In fact, it is enough to evaluate the effective support size Ω, the number of different sequences that P can generate.

Ω can be computed as the exponential of the entropy S of the associated probability distribution $P(a_1, \ldots, a_L)$:

$$\Omega = \exp(S) \tag{4.1}$$

$$S = -\sum_{a_1,\ldots,a_L} P(a_1, \ldots, a_L) \log P(a_1, \ldots, a_L) \tag{4.2}$$

In a DCA model $P(a_1, \ldots, a_L)$ a very hard problem is to compute the value

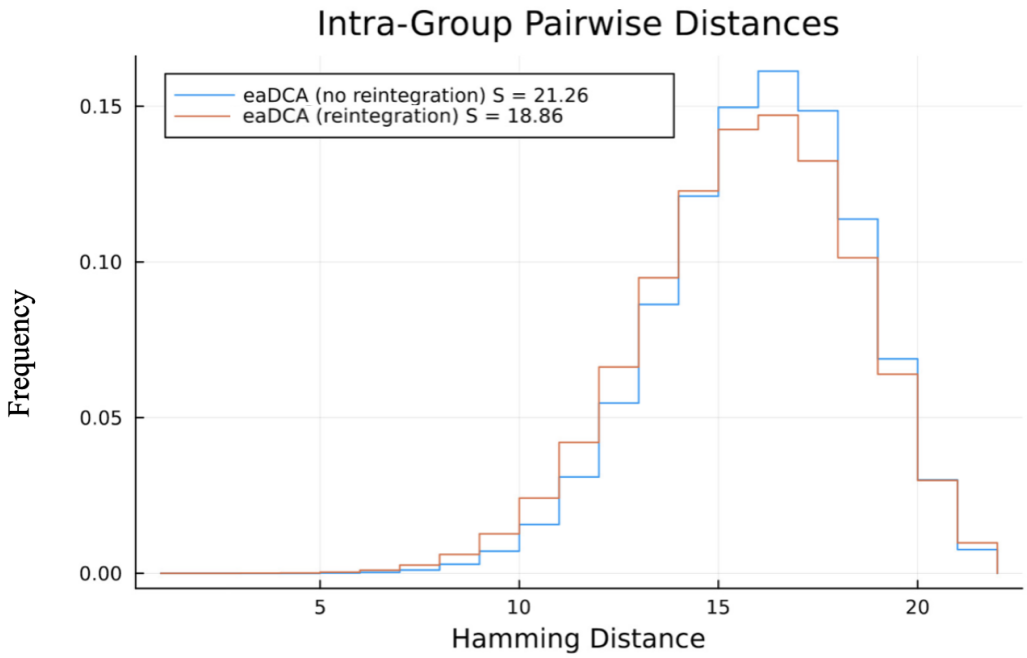

FIGURE 4.6: Histogram of the intra-group pairwise distances computed between the sequences before reintegration and after reintegration

of the partition function Z. Once the value of Z is known, it becomes straightforward to compute the model entropy S :

$$S = -<\log P(a_1, \ldots, a_L)>_P \tag{4.3}$$

With eaDCA it is possible to monitor the partition function Z during the training. Remembering eq. 2.11, the entropy of the model can be computed as :

$$S = <E_t>_{P_t} \tag{4.4}$$

Even though this relation is not really exact due to regularization and MCMC sampling, this way of computing the entropy is enough to estimate the diversity of our model.
Another estimate of diversity is computing the intra-group pairwise distances between the sampled artificial sequences. This is an estimate of how different are sequences in a sample: for each couple of sequence in the sample, the hamming distance between them is computed and collected. After this an histogram is plotted and the broader is the histogram, the more diverse is the sample.
Both the entropy test and the distances test were performed for the hammerhead ribozyme and the histogram of the intra-group pairwise distances is plotted in figure 4.5 both for the sample from the model before reintegration and for the sample from the model after reintegration. The plot shows that there is no appreciable change in terms of diversity.
Furthermore the entropy of the two models was computed and the reduction is very small, going from $S = 21.26$ for the model before reintegration to $S = 18.86$ for the model after reintegration. Remembering eq. 4.1, the size of Ω is reduced of a factor ~ 11 which is really a small reduction when looking at the thermo scores in Figure 4.4: our models do not have diversity problems in this scenario.

Chapter 5

Reintegration: Real Data

5.1 eaDCA for the Azoarcus Setting

After the success with the tests on synthetic data, our reintegration methods were applied to real data belonging to the Azoarcus setting, described in Chapter 1.
Let us start again from the question: Is RNA reproduction widespread in the sequence space?
Azoarcus bacterium's group I intron ribozyme is one of the few known self-replicant RNA. It is fundamental to find other self-replicant RNAs in order to enhance the plausibility of the RNA world theory [8].
eaDCA is well suited for the job, since it is capable of estimating the order of magnitude number of potential functional sequences within a given RNA family [12].
In recent years Azoarcus setting was studied in depth [27] and in the following there will be a review of currently unpublished data and methods developed at ESPCI Paris that will be very helpful for the application of our reintegration methods to real data.
As the name suggests, Azoarcus RNA belongs to the family of "group I intron ribozymes". In its studying process, the alignment of such family was based on Azoarcus RNA and it is of poor quality since it is very gapped and also it is trimmed to Azoarcus length. Anyway, from this alignment it was possible to learn a DCA model H_A via edge activation.
This model is expected to be of low quality due to the alignment based on Azoarcus. So sampling sequences independently would lead to non-functional sequences. However, it is possible bias the sampling to an environment of the Azoarcus sequence, hoping to find mutated but functional sequences, i.e. self-replicator candidates.
These model-informed mutations from Azoarcus RNA can be generated thanks to Importance Sampling (Ch.2).
Usual Gibbs Sampling is not the solution here: to generate mutations close to the wild-type one could think of generating a big sample and filter the mutations at desired distance. This method is very inefficient since it was observed that the sequences generated via Gibbs sampling have a distance from the wild-type which is distributed as a rapidly decaying distribution peaked from Azoarcus RNA.
This is shown in Figure 5.1.

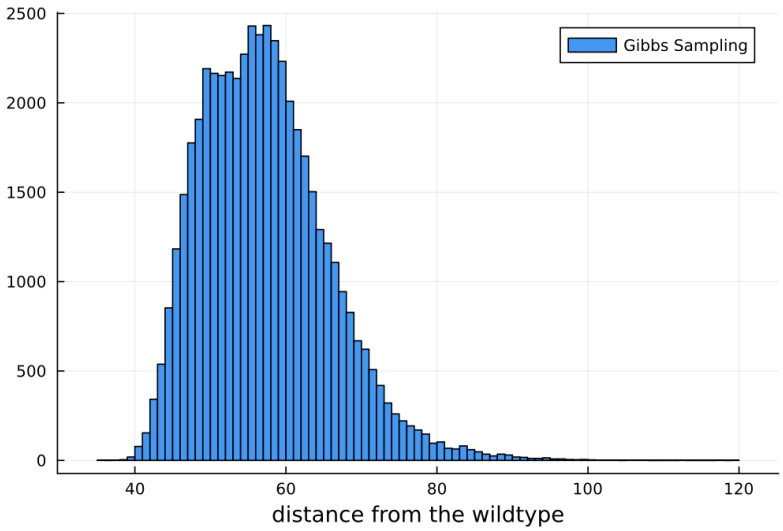

FIGURE 5.1: Histogram of the distances from Azoarcus RNA of the sequences generated via usual Gibbs Sampling from the original eaDCA model H_A.

For this reason one would need an astronomically large sample in order to get a good amount of mutations closer to Azoarcus RNA.
Importance Sampling is used to generate efficiently mutations from the wild-type. It is enough to change the fields of our Potts model H_A by introducing a sampling strength θ, according to eq. 2.24.
In this way the model assigns more probability to the nucleotides present in Azoarcus (wild-type) but sometimes mutations based on the model can occur during Gibbs Sampling.
By opportunely tuning the value of θ, it is possible to sample mutations at every distance from the wild-type.
Then these artificial sequences are tested in bulk for self-splicing with high-throughput experiments and, simplifying, a catalytic activity is measured for each sequence.
Since this activity is a good proxy for self-replication, artificial sequences with an activity higher than some threshold are good candidates and they can be tested for self-replication in a low-throughput experiment.
So the process in short is : start from the alignment of the "group I intron ribozymes" and learn an eaDCA model. Then generate, via Importance Sampling, mutations at every distance from the wildtype and mass-test them.
A representation of this process is drawn in Figure 5.2.
An important insight on the artificial sequences is the active fraction. It is defined as the percentage of generated sequences, at fixed distance from the wild-type, that have activity higher than a threshold.
In figure 5.3 there is an example of "active fraction vs distance" plot. In this case, starting from the eaDCA model, bins of 400 sequences were generated via Importance Sampling at each distance multiple of 5 (from 5 to 75). Then their activity was measured experimentally and a threshold was set: setting the threshold is a problem itself.

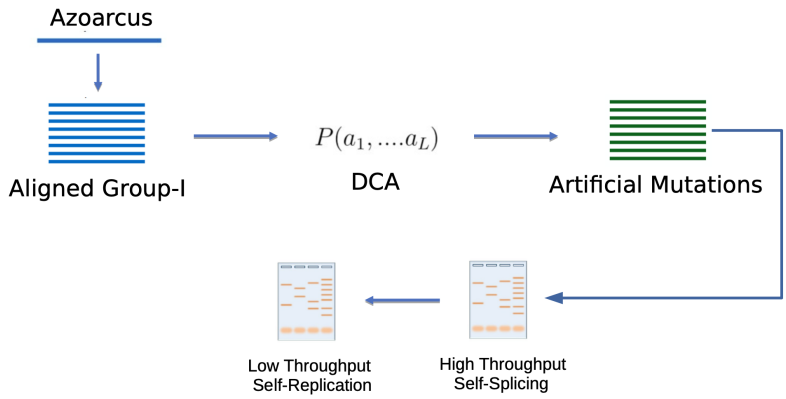

FIGURE 5.2: Schematics of the setting of eaDCA applied to the Azoarcus setting.

Then the active fraction was computed for each bin. For example, if in the

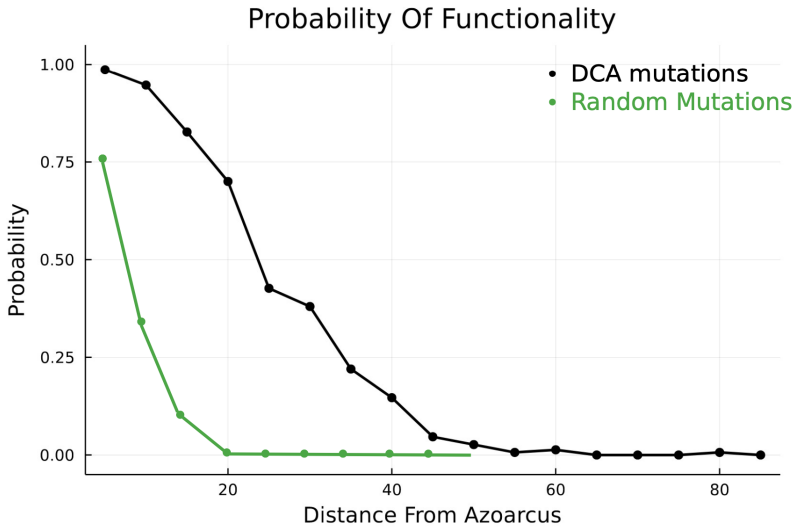

FIGURE 5.3: Active fraction of the mutations generated via Importance Sampling from eaDCA model H_A compared to the active fraction of randomly generated mutations.

20-distance bin there are 280 sequences with activity above the threshold and 120 below, the active fraction would be of 0.7.

In parallel, the active fraction was also computed for sequences generated with random mutations. Here we can see the importance of the model : already at distance 20 the active fraction is 0 for the random sequences. Instead, with the eaDCA model H_A it is possible to find a non-zero active fraction even at distance 60.

Actually a lot more can be done via eaDCA. Thanks to the possibility of computing the entropy of the model (and consequently the size of the model support) and thanks to the possibility of computing a rate of functionality, the number of potential self-replicants can be estimated. It is very huge, with this estimate order 10^{40} sequences are deemed as potential self-replicants, as

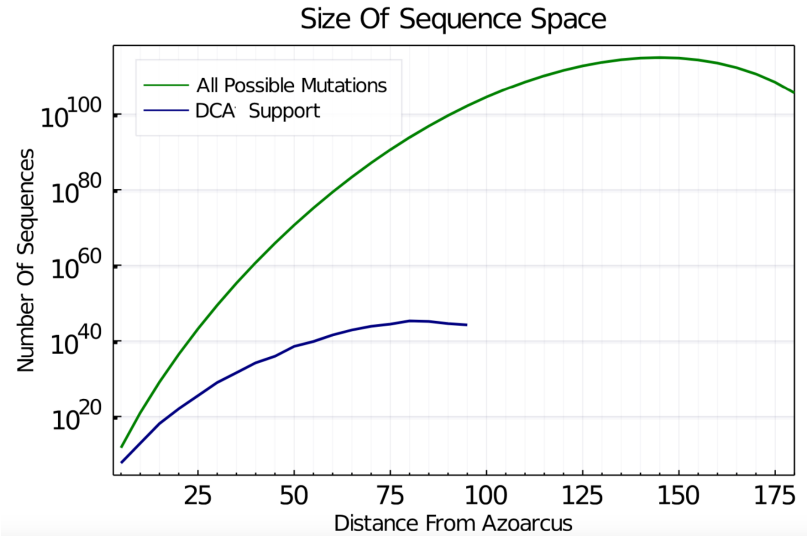

FIGURE 5.4: Estimation of the DCA support compared to that of all possible mutations.

shown in Figure 5.4.
This is what was done by others : here the review finishes and the goal of this chapter is to start from these results and apply our reintegration methods to learn a new refined model.
More precisely we started from an eaDCA model H_A learnt on "group I intron ribozymes" and from a dataset D_A composed of about 26000 artificial mutations whose activity was measured experimentally.
In figure 5.5 it is represented the scatter plot between distance and activity of sequences in dataset D_A. Looking at it, a good threshold value could be set at activity -2.5 since after distance 70 almost no sequence is functional : those which seem functional are actually a product of experimental noise.
These sequences came from different models: DCA, Variational Autoen-

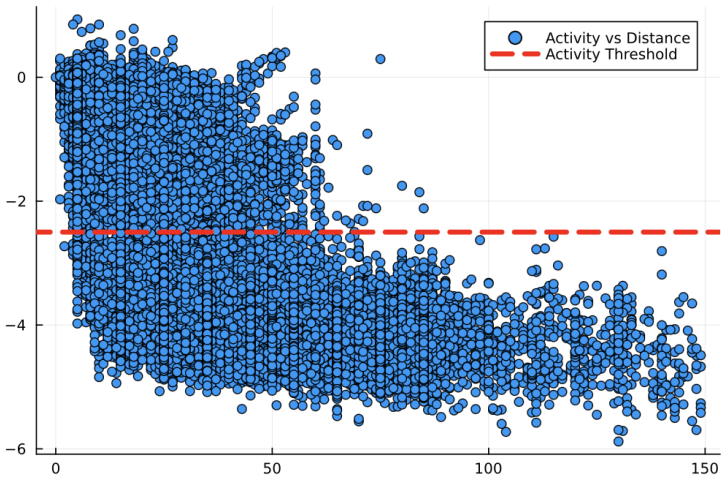

FIGURE 5.5: Scatter plot between distance and activity of sequences in dataset D_A : the threshold value is pictured in red

coders [28] and Thermodynamic Models [29].

To apply our methods though, some issues need to be addressed.

5.2 Global Reintegration and Local Reintegration

Let us start with a brief summary about the reintegration methods developed in Chapter 3, showing parallelisms and differences with respect to our setting here.
The reintegration methods started from a classical DCA model and its likelihood 2.3. Here the scenario is the same since the eaDCA model H_A learnt on the "group I intron ribozymes" is given.
From the DCA likelihood a new objective function 3.12. is designed to include the information coming from the annotated sequences. The key role is played by the adjustment function. Based on the score of a sequence its effect is to decrease the probability assigned to bad sequences and viceversa increase the probability assigned to good sequences.
In this way new target frequencies are computed, a new model can be learnt and artificial sequences can be sampled from it.
Until now the reintegration we did can be defined as "global". The tested sequences were always an iid sample obtained via Gibbs Sampling from the model that we wanted to improve.
In the Azoarcus setting instead the data that we want to reintegrate are all mutations of the wild-type generated via Importance Sampling. Again, this is due to the fact that the alignment based on Azoarcus is very gapped.
Then the experimentally tested dataset is not anymore an iid sample and the reintegration can be defined as "local".
While the global reintegration has proven to be successful as it is, the local reintegration is more complicate and it requires some attention.
One could be tempted of reintegrating the sequences in the dataset D_A all at once, giving adjustment function +1 to all the sequences with activity above the threshold and -1 to the ones with activity below the threshold.
Unfortunately this naive type of reintegration brings a strong bias towards the wild-type. All the reintegrated sequences are mutations of the Azoarcus RNA so we are not reintegrating an iid sample. Furthermore mutations at bigger distance have smaller activity so, when adjusting the frequencies, all the useful signal is cancelled by this bias. The only thing that our reintegrated model learns is that sequences near to the wild-type are good and, vice versa, sequences far away from the wild-type are bad.

As shown qualitatively in Figure 5.6, our reintegrated model is far more concentrated around the wild-type. In the next paragraph we will see how to get rid of this problem.

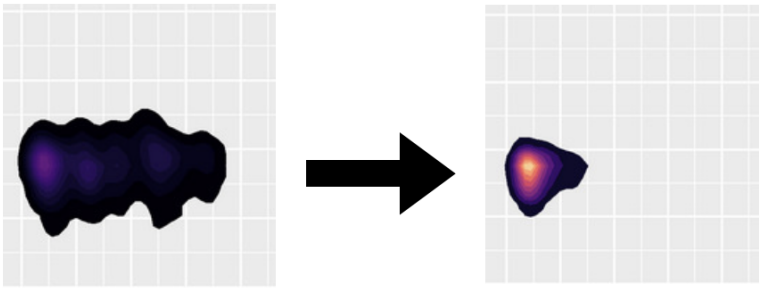

FIGURE 5.6: A qualitative representation of the bias introduced when reintegrating naively : on the left we have a measure of how spread is the original model, on the right we have the reintegrated model, very concentrated on the wild-type.

5.3 Solutions to the Bias Problems

There are probably many ways to get rid of the bias previously described. We would want to cancel out the strong signal towards the bias so that we can catch the useful information embedded in the tested sequences.
Before proposing a solution, let us recap the setting. The starting point is an eaDCA model and a big dataset of tested mutations.
These mutations are labelled as good or bad depending on how their activity compares with the threshold. So in the following we will refer to "good" sequences if their activity is bigger than -2.5 and to "bad" sequences if their activity is lower that -2.5.
The crucial point to get rid of the bias is that the sequences in D_A can be grouped in bins. The binning size is actually an hyper-parameter of the problem and it will be addressed in a moment. For each bin there is a number of good sequences N_+ (activity greater than threshold) and a number of bad sequences N_-.
This is shown in Figure 5.8 for a 4 binning size.
For each bin is also reported the correlation by mutation bin: it is the correlation between distance and activity of sequences in a given bin. The binning size should be designed so that there is no appreciable correlation by mutation bin.
Our strategy is the following : the adjustment function w is designed so that

$$\sum_{\vec{b} \in bin} w(\vec{b}) = 0$$

At each distance the signal towards the wild-type is cancelled out when adjusting the frequencies.
A way to do this is to take $w(\vec{b}) = \frac{1}{N_+}$ for every $\vec{b} \in bin$ that is a good sequences and $w(\vec{b}) = -\frac{1}{N_-}$ for every $\vec{b} \in bin$ that is a bad sequence. This is shown in Figure 5.9. Now there could be an issue: if in a bin there are very

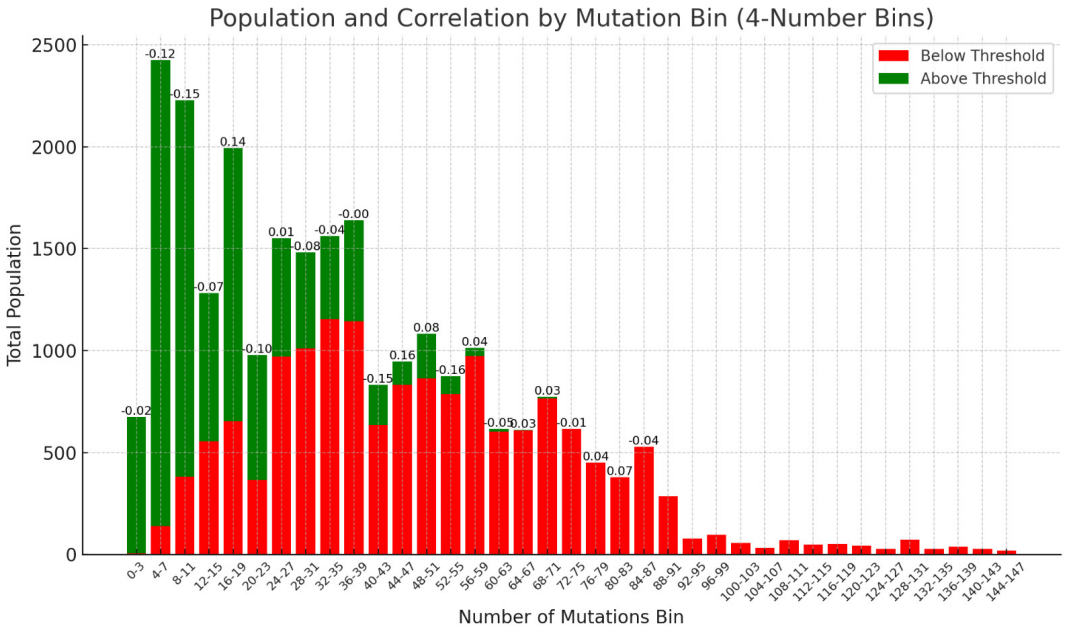

FIGURE 5.7: Representation of the Population of bins (good and bad sequences) with also the in-bin Correlation between activity and distance of sequences in a given bin.

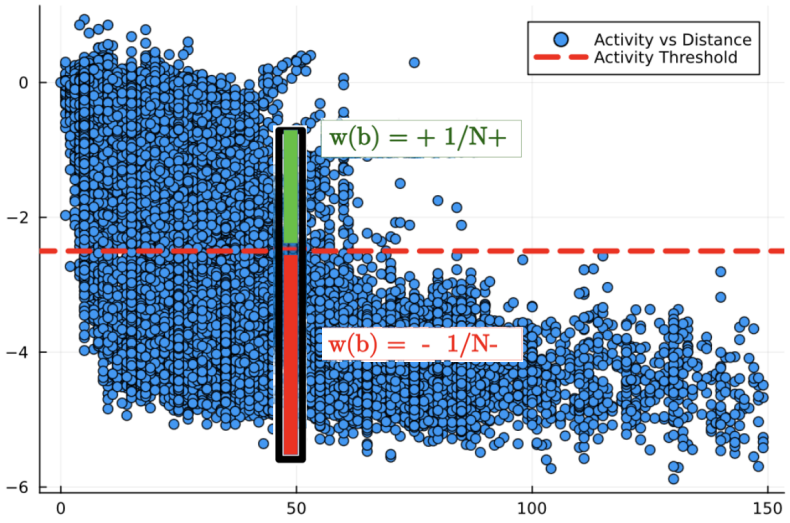

FIGURE 5.8: Example of how the adjustment function is designed for a single bin

few bad sequences they could receive a very big negative adjustment function: this is a big problem since it could bring some convergence problems.
At the same time, if in a bin there are few good sequences and many bad sequences, the positive ones would get a very big reintegration weight. This is not a big problem but it could direct too much the frequencies and maybe this few sequences are product of experimental noise.
For this reason only bins in which there are a good amount of both good and bad sequences are taken into account.

Since we still want to reintegrate as many sequences as possible here the binning size can be engineered in order to include a bigger number of sequences.

5.4 Reintegration Methods on Azoarcus: Results

Starting from the experimental dataset divided in bins, it is possible to apply our reintegration methods to the original eaDCA model H_1.
After some analysis we decided to divide the tested sequences in bins of width 4 and reintegrate mutations from distance 4 to 60. For each bin the adjustment for each sequence is assigned with the rule described before based on the activity value of the specific sequence.
In this way the new target frequencies are computed thanks to eqs. 3.13 and 3.14. Some more analysis was done to find the optimal reintegration strength λ.
If the average adjustment $w(\vec{b})$ is of order 0.001, we found that a good value for the reintegration strength is $\lambda = 100 - 150$ so that the reintegration affects not too much the training and convergence is guaranteed.
Then a new model H_R is trained via eaDCA using these new target frequencies. Once the model is trained we can use importance sampling to generate bins of 150 artificial mutations at distance ranging from 5 to 75. This was done for both the original model H_1 and the reintegrated model H_R for a total of $2 \cdot 150 \cdot 15 = 4500$ sequences.

How can we access if these sequences are better than those sampled from

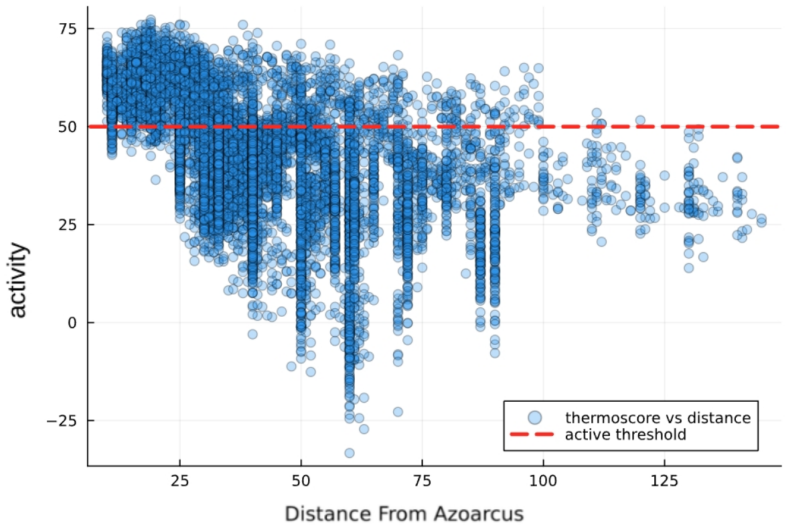

FIGURE 5.9: Scatter plot between distance and ViennaRNA package thermo score of most sequences in dataset D_A : the threshold value is pictured in red

the first DCA model ? The best way are obviously new experiments. To prevalidate our results we have used the ViennaRNA Thermo score as a computational fitness proxy. The secondary structure of Azoarcus RNA is known

and it can be written in a dot-bracket representation. Then all the artificial mutations can be tested by computing, via RNAEval, their Thermo Scores ($-$ thermodynamic Free Energy) of folding into that structure.
To compute the thermo-stable fraction at a given distance, it is enough to set a Thermo Score threshold. This is done again looking at the reintegration dataset: since some of these mutations came from a model based on secondary structure, they are omitted for this purpose.
Looking at the scatter plot in Figure 5.9, we see that the thermo score behaves similarly to the experimental active fraction, decaying with distance. A threshold close to $TS = 50$ seems reasonable when comparing to Fig. 5.5.
Now the thermo-stable fraction is easily computed and it is possible to validate the mutations generated from our models. The result for both model is drawn in Figure 5.10.
At given distance, the thermo-stable fraction of mutations sampled from the reintegrated model is far higher than the active fraction of mutations sampled from the original model.
The reintegration was done based on activity so we expect that testing the experimental activity of mutations will lead to even better results in terms of active function.
With these results, we have convinced our experimental colleagues at ESPCI Paris to test sequences generated from the reintegrated model. While the tests will be possible only after the end of this master project, they will be the definite "real life" check of the proposed method.

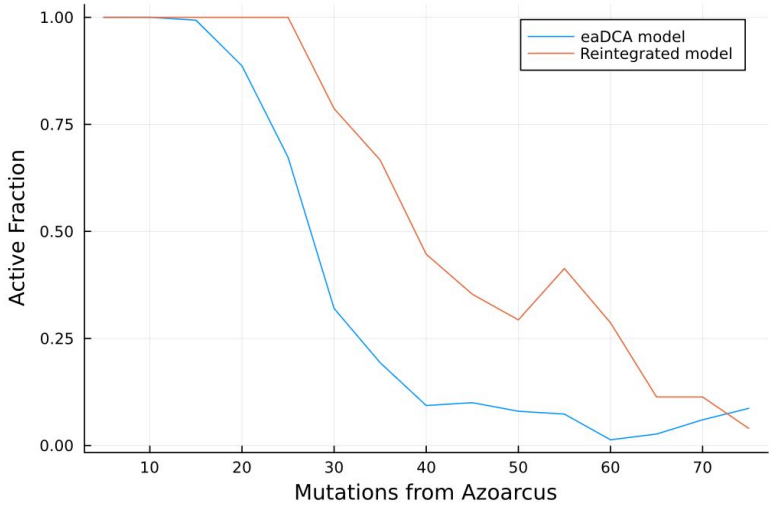

FIGURE 5.10: Active fraction based on ViennaRNA thermo score of artificially generated mutations from both the original and the reintegrated model

Chapter 6

Conclusions

The goal of this thesis was to develop methods to integrate annotated data into existent generative models of biomolecules.
Starting with an analytical derivation we succeeded in finding a relatable strategy to do so. Then tests on both synthetic and real data were done with promising results.
In the near future new experiments are scheduled at ESPCI Paris. They will create the mutations generated from our reintegrated model and they will test their activity as well. Our hope is that those mutations perform better in terms of active fraction at fixed distance. On top of that we hope that we are able to find some active mutations even at greater distance with respect to existent models. If the first result would be already great, the second would be of maximal importance for the work that they are doing at ESPCI: it would increase the order of magnitude of the entropy-estimated number of self-replicators candidates, giving further strength to RNA world theory.
When learning a statistical model there can be at least two limits : the model and the data.
Even though lately the main direction is to use more and more complicated models (as Large Language Models), in this thesis we kept an already existing model and we instead focused on the data since lately there is an increasing possibility to do experiments.
This yielded good results and restricting the focus on experimental information should be something more adopted. High-throughput experiments can be done more easily and therefore reintegration methods using “a la carte” data seem a relevant direction: the hope is that doing cycles of experiments after the training of statistical models can improve them greatly.
This a good modus operandi in the sequence design problem and it is not restricted to DCA models.

Bibliography

[1] Dror Baran et al. "Principles for computational design of binding antibodies". In: *Proceedings of the National Academy of Sciences* 114.41 (2017), pp. 10900–10905. ISSN: 0027-8424. DOI: 10.1073/pnas.1707171114. eprint: https://www.pnas.org/content/114/41/10900.full.pdf. URL: https://www.pnas.org/content/114/41/10900.

[2] William P. Russ et al. "An evolution-based model for designing chorismate mutase enzymes". In: *Science* 369.6502 (2020), pp. 440–445. ISSN: 0036-8075. DOI: 10.1126/science.aba3304. eprint: https://science.sciencemag.org/content/369/6502/440.full.pdf. URL: https://science.sciencemag.org/content/369/6502/440.

[3] S. Cocco C. Feinauer M. Figliuzzi R. Monasson and M. Weigt. "Inverse statistical physics of protein sequences: a key issues review". eng. In: *Reports on Progress in Physics* (2018). DOI: https://doi.org/10.1088/1361-6633/aa996. URL: https://arxiv.org/abs/1703.01222.

[4] Lee E. Vandivier et al. "The Conservation and Function of RNA Secondary Structure in Plants". eng. In: *Annual review of plant biology* 67 (Apr. 2016). PMC5125251[pmcid], pp. 463–488. ISSN: 1545-2123. DOI: 10.1146/annurev-arplant-043015-111754. URL: https://doi.org/10.1146/annurev-arplant-043015-111754.

[5] Sam Griffiths-Jones et al. "Rfam: an RNA family database". eng. In: *Nucleic acids research* 31.1 (Jan. 2003). PMC165453[pmcid], pp. 439–441. ISSN: 1362-4962. DOI: 10.1093/nar/gkg006. URL: https://doi.org/10.1093/nar/gkg006.

[6] E. P. Nawrocki and S. R. Eddy. "Infernal 1.1: 100-fold faster RNA homology searches". eng. In: *Bioinformatics* (2013). DOI: https://doi.org/10.1093/bioinformatics/btt509. URL: https://academic.oup.com/bioinformatics/article/29/22/2933/316439.

[7] James W. Brown et al. "The RNA structure alignment ontology". eng. In: *RNA (New York, N.Y.)* 15.9 (Sept. 2009). rna.1601409[PII], pp. 1623–1631. ISSN: 1469-9001. DOI: 10.1261/rna.1601409. URL: https://doi.org/10.1261/rna.1601409.

[8] Michael P. Robertson and Gerald F. Joyce. "The origins of the RNA world". eng. In: *Cold Spring Harbor perspectives in biology* 4.5 (May 2012). cshperspect.a003608[PII], a003608. ISSN: 1943-0264. DOI: 10.1101/cshperspect.a003608. URL: https://doi.org/10.1101/cshperspect.a003608.

[9] P. Pavlinova C. N. Lambert C. Malaterre and P. Nghe. "Abiogenesis through gradual evolution of autocatalysis into template-based replication". eng. In: *Febs Letters* (2022). DOI: https://doi.org/10.1002/1873-3468.14507. URL: https://febs.onlinelibrary.wiley.com/doi/10.1002/1873-3468.14507.

[10] S. Mukherjee D. Stamatis J. Bertsch N.C. Kyrpides et al. "Twenty-five years of Genomes OnLine Database (GOLD): data updates and new features in v.9". eng. In: *Nucleic Acids Research* (2023). DOI: https://doi.org/10.1093/nar/gkac974.

[11] R. Durbin S. R. Eddy A. Krogh and G. Mitchison. *Biological sequence analysis: probabilistic models of proteins and nucleic acids*. Cambridge University Press, 1998. URL: https://books.google.it/books/about/Biological_Sequence_Analysis.html?id=R5P2GlJvigQC&redir_esc=y.

[12] F. Zamponi M. Weigt F. Calvanese C. N. Lambert P. Nghe. "Towards Parsimonious Generative Modeling of RNA Families". eng. In: (2024). DOI: https://doi.org/10.1101/2023.10.19.562525. URL: https://arxiv.org/abs/2310.12700.

[13] Matteo Figliuzzi Pierre Barrat-Charlaix Martin Weigt. "How Pairwise Coevolutionary Models Capture the Collective Residue Variability in Proteins?" In: *bioRxiv* (2018). DOI: https://doi.org/10.1093/molbev/msy007. URL: https://academic.oup.com/mbe/article/35/4/1018/4815777.

[14] Thomas Gueudré et al. "Simultaneous identification of specifically interacting paralogs and interprotein contacts by direct coupling analysis". In: *Proceedings of the National Academy of Sciences* 113.43 (2016), pp. 12186–12191. ISSN: 0027-8424. DOI: 10.1073/pnas.1607570113. eprint: https://www.pnas.org/content/113/43/12186.full.pdf. URL: https://www.pnas.org/content/113/43/12186.

[15] J. Trinquier G. Uguzzoni A. Pagnani F. Zamponi and M. Weigt. "Efficient generative modeling of protein sequences using simple autoregressive models". eng. In: *Nature Communications* (2021). DOI: https://doi.org/10.1038/s41467-021-25756-4. URL: https://www.nature.com/articles/s41467-021-25756-4#citeas.

[16] H. Chau Nguyen Riccardo Zecchina Johannes Berg. "Inverse statistical problems: from the inverse Ising problem to data science". eng. In: *Advances in Physics* (2017). DOI: https://doi.org/10.48550/arXiv.1702.01522. URL: https://arxiv.org/abs/1702.01522.

[17] M. Weigt R. A. White H. Szurmant J. A. Hoch. "Identification of direct residue contacts in protein-protein interaction by message passing". eng. In: *Proceedings of the National Academy of Sciences* (2009). DOI: https://doi.org/10.1073/pnas.080592310. URL: https://www.pnas.org/doi/full/10.1073/pnas.0805923106.

[18] F. Morcos A. Pagnani B. Lunt A. Bertolino D. S. Marks C. Sander R. Zecchina J. N. Onuchic T. Hwa and M. Weigt. "Direct-coupling analysis of residue coevolution captures native contacts across many protein families". eng. In: *Proceedings of the National Academy of Sciences* (2011). DOI: https://doi.org/10.1073/pnas.11114711. URL: https://www.pnas.org/doi/10.1073/pnas.1111471108.

[19] S. Panzeri, C. Magri, and L. Carraro. "Sampling bias". In: *Scholarpedia* 3.9 (2008). revision #148550, p. 4258. DOI: 10.4249/scholarpedia.4258.

[20] J. Felsenstein. "Inferring Phylogenies". eng. In: *Sinauer Associates, Sunderland, Massachusetts* (2003). DOI: https://doi.org/10.1080/10635150490468530.

[21] E. T. Jaynes. *Probability Theory: The Logic of Science*. Cambridge University Press, 2003. ISBN: 9780521592710. URL: https://books.google.it/books/about/Probability_Theory.html?id=tTN4HuUNXjgC&redir_esc=y.

[22] D. H. Ackley G. E. Hinton and T. J. Sejnowski. "A Learning Algorithm for Boltzmann Machines". eng. In: *Cognitive Science* (1985). URL: https://www.sciencedirect.com/science/article/abs/pii/S0364021385800124.

[23] Adrian Barbu and Song-Chun Zhu. *Monte Carlo Methods*. Springer Singapore, 2020. DOI: 10.1007/978-981-13-2971-5. URL: https://doi.org/10.1007/978-981-13-2971-5.

[24] Lorenz R. Bernhart S.H. Höner zu Siederdissen C. et al. "ViennaRNA Package 2.0". eng. In: *Algorithms for Molecular Biology* (2011). DOI: https://doi.org/10.1186/1748-7188-6-26. URL: https://almob.biomedcentral.com/articles/10.1186/1748-7188-6-26.

[25] M. Zuker and P. Stiegler. "Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information". eng. In: *Nucleic Acids Research* (1981). DOI: https://doi.org/10.1093/nar/9.1.133. URL: https://academic.oup.com/nar/article/9/1/133/1043226.

[26] P. Barrat-Charlaix M. Figliuzzi and M. Weigt. "Improving landscape inference by integrating heterogeneous data in the inverse Ising problem". eng. In: *Scientific Reports* (2016). DOI: https://doi.org/10.1038/srep37812. URL: https://www.nature.com/articles/srep37812.

[27] C. Jeancolas C. N. Lambert P. Nghe et al. "RNA diversification by a self-reproducing ribozyme revealed by deep sequencing and kinetic modelling". eng. In: *Chemical Communications* (2021). DOI: https://doi.org/10.1039/D1CC02290C.

[28] D. P. Kingma M. Welling. "An Introduction to Variational Autoencoders". eng. In: *Foundations and Trends in Machine Learning* (2019). DOI: https://doi.org/10.48550/arXiv.1906.02691.

[29] D. H Mathews M. D Disney J. L. Childs S. J. Schroeder M. Zuker D. H. Turner. “Incorporating chemical modification constraints into a dynamic programming algorithm for prediction of RNA secondary structure”. eng. In: *PNAS* (2004). DOI: https://doi.org/10.1073/pnas.0401799101.